CE706 - Information Retrieval 2021

Assigment 1

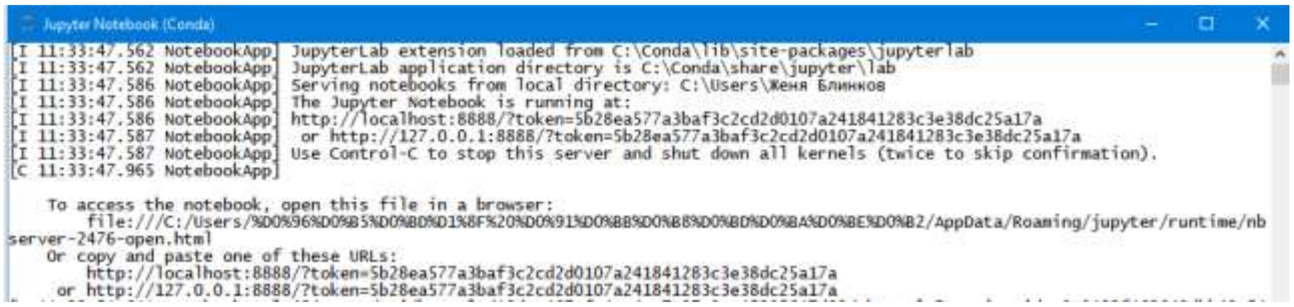
Student ID: 2003667

Instructions for running your system (Engineering a Complete System)

Include here instructions to run your system and control each individual component. You may include screenshots to clarify.

To run the system a user should have already installed Jupyter Notebook, ElasticSearch and Kibana.

First of all, a user should launch a Juper Notebook server using Jupyter Notebook (Conda). In the result, a user will see a window with similar information as it is shown on the figure 1.

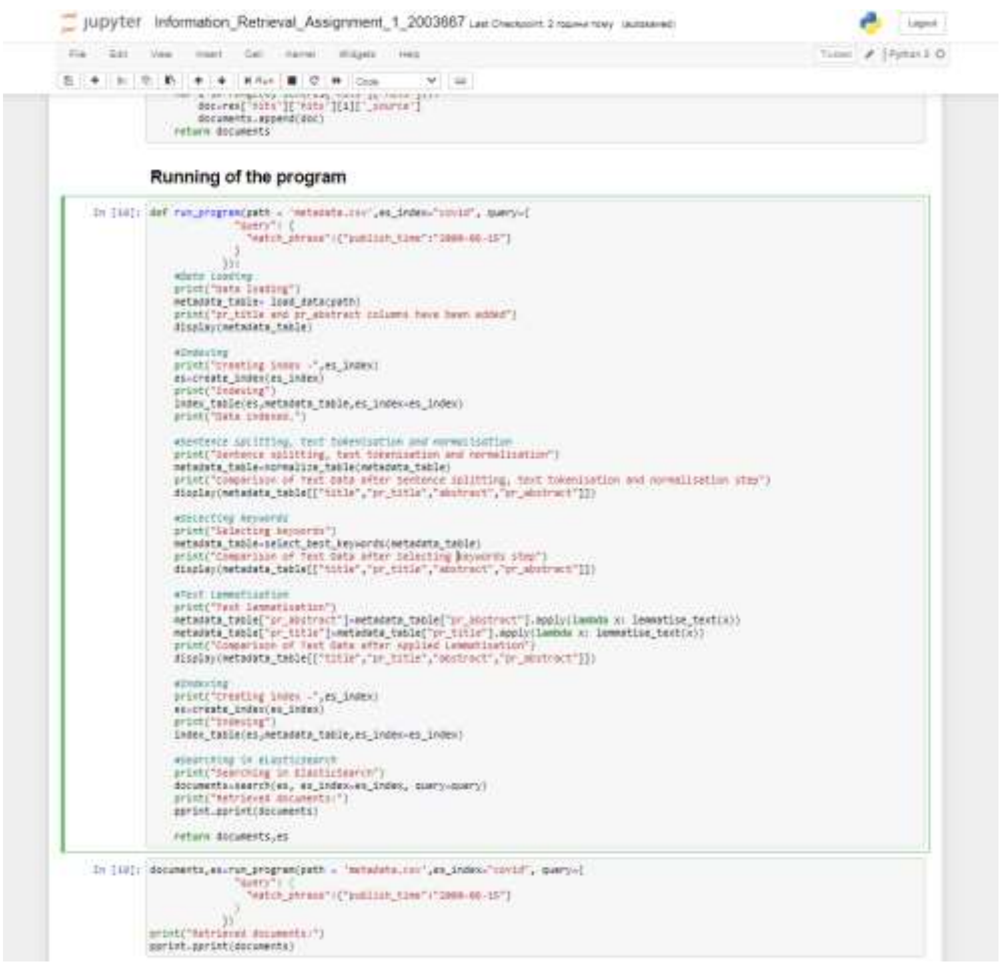


```
[I 11:33:47.562 NotebookApp] JupyterLab extension loaded from C:\Conda\lib\site-packages\jupyterlab
[I 11:33:47.562 NotebookApp] JupyterLab application directory is C:\Conda\share\jupyter\lab
[I 11:33:47.586 NotebookApp] Serving notebooks from local directory: C:\Users\Женя Блинков
[I 11:33:47.586 NotebookApp] The Jupyter Notebook is running at:
[I 11:33:47.586 NotebookApp] http://localhost:8888/?token=5b28ea577a3baf3c2cd2d0107a241841283c3e38dc25a17a
[I 11:33:47.587 NotebookApp]  or http://127.0.0.1:8888/?token=5b28ea577a3baf3c2cd2d0107a241841283c3e38dc25a17a
[I 11:33:47.587 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 11:33:47.965 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/%D0%96%D0%B5%D0%BD%D1%8F%20%D0%91%D0%BB%D0%B8%D0%BD%D0%BA%D0%BE%D0%B2/AppData/Roaming/jupyter/runtime/nb
server-2476-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=5b28ea577a3baf3c2cd2d0107a241841283c3e38dc25a17a
     or http://127.0.0.1:8888/?token=5b28ea577a3baf3c2cd2d0107a241841283c3e38dc25a17a
```

Figure 1: Screenshot of a Jupyter Notebook server launching window.

Having copied and pasted one of those URLs from the figure 2 into a web-browser, a user should open an Information_Retrieval_Assignment_1_2003667.ipynb file. In the result, the following web-page, that is shown on the figure 2, will be opened.

Although, alternatively, if a user does not have a Jupyter Notebook, he can open it on Google Collaboratory, or just open a provided .py file. In the last case, a further sequence of actions with the programme remains the same


jupyter Information_Retrieval_Assignment_1_2003867 Last Checkpoint: 2 години тому (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code

```
        doc=res["hits"]["hits"][i]["_source"]
        documents.append(doc)
    return documents
```

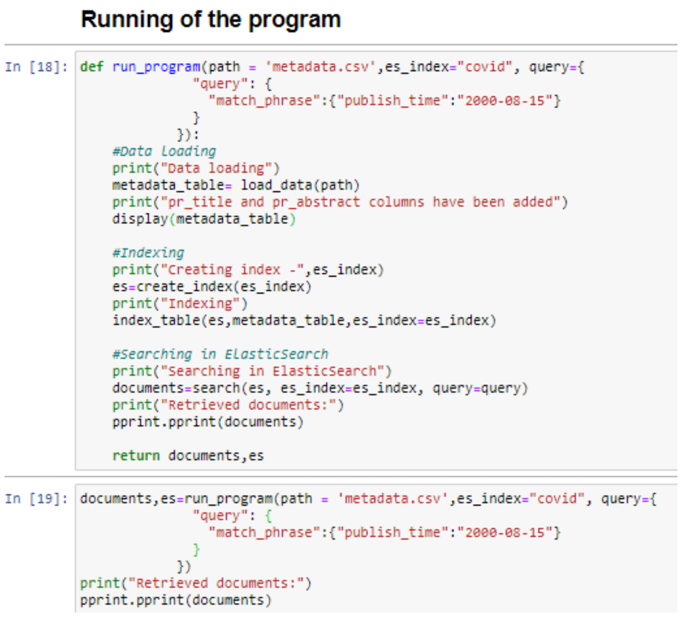
Running of the program

```
In [18]: def run_program(path = 'metadata.csv',es_index="covid", query={
                    "query": {
                      "match_phrase":{"publish_time":"2000-08-15"}
                    }
                  }):
             #Data Loading
             print("Data loading")
             metadata_table= load_data(path)
             print("pr_title and pr_abstract columns have been added")
             display(metadata_table)

             #Indexing
             print("Creating index -",es_index)
             es=create_index(es_index)
             print("Indexing")
             index_table(es,metadata_table,es_index=es_index)
             print("Data indexed.")

             #Sentence splitting, text tokenisation and normalisation
             print("Sentence splitting, text tokenisation and normalisation")
             metadata_table=normalize_table(metadata_table)
             print("Comparison of Text data after sentence splitting, text tokenisation and normalisation step")
             display(metadata_table[["title","pr_title","abstract","pr_abstract"]])

             #Selecting keywords
             print("Selecting keywords")
             metadata_table=select_best_keywords(metadata_table)
             print("Comparison of Text data after Selecting keywords step")
             display(metadata_table[["title","pr_title","abstract","pr_abstract"]])

             #Text Lemmatisation
             print("Text lemmatisation")
             metadata_table["pr_abstract"]=metadata_table["pr_abstract"].apply(lambda x: lemmatise_text(x))
             metadata_table["pr_title"]=metadata_table["pr_title"].apply(lambda x: lemmatise_text(x))
             print("Comparison of Text data after applied Lemmatisation")
             display(metadata_table[["title","pr_title","abstract","pr_abstract"]])

             #Indexing
             print("Creating index -",es_index)
             es=create_index(es_index)
             print("Indexing")
             index_table(es,metadata_table,es_index=es_index)

             #Searching in ElasticSearch
             print("Searching in ElasticSearch")
             documents=search(es, es_index=es_index, query=query)
             print("Retrieved documents:")
             pprint.pprint(documents)

             return documents,es
```

```
In [19]: documents,es=run_program(path = 'metadata.csv',es_index="covid", query={
                    "query": {
                      "match_phrase":{"publish_time":"2000-08-15"}
                    }
                  })
         print("Retrieved documents:")
         pprint.pprint(documents)
```

Figure 2: Screenshot of the program-running part of the code for the assignment 1.

According to the figure above, a user can choose which components he would like to run by including or excluding certain parts of the code in the function “run_program”. An example of a desired sequence of steps can be seen on the figure 3 below.



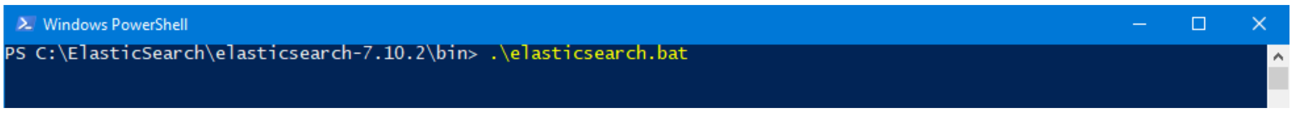
Running of the program

```
In [18]: def run_program(path = 'metadata.csv',es_index="covid", query={
                        "query": {
                          "match_phrase":{"publish_time":"2000-08-15"}
                        }
                      }):
             #Data Loading
             print("Data loading")
             metadata_table= load_data(path)
             print("pr_title and pr_abstract columns have been added")
             display(metadata_table)

             #Indexing
             print("Creating index -",es_index)
             es=create_index(es_index)
             print("Indexing")
             index_table(es,metadata_table,es_index=es_index)

             #Searching in ElasticSearch
             print("Searching in ElasticSearch")
             documents=search(es, es_index=es_index, query=query)
             print("Retrieved documents:")
             pprint.pprint(documents)

             return documents,es
```

```
In [19]: documents,es=run_program(path = 'metadata.csv',es_index="covid", query={
                        "query": {
                          "match_phrase":{"publish_time":"2000-08-15"}
                        }
                      })
         print("Retrieved documents:")
         pprint.pprint(documents)
```

Figure 3: Screenshot of the program-running part of the code for the assignment 1 with excluded preprocessing steps.

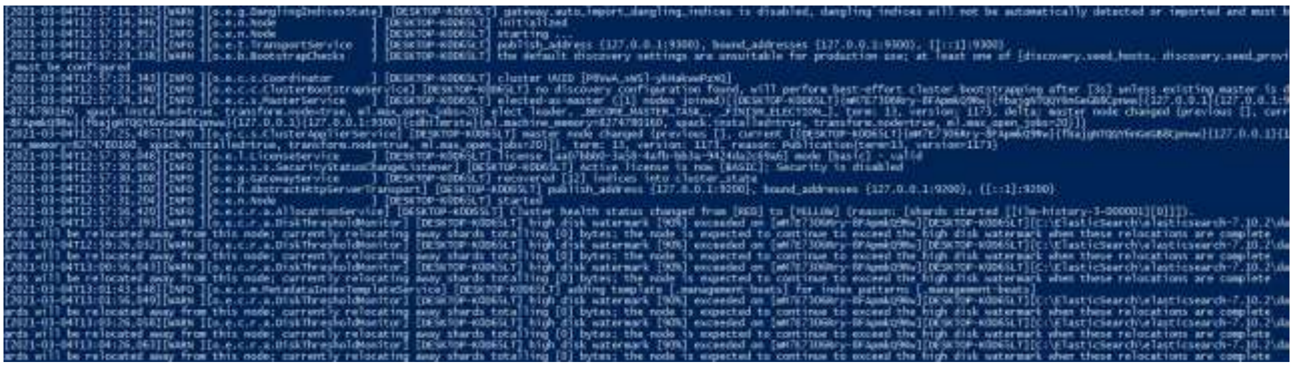
Before running the program, a user should open a command line window to launch ElasticSearch engine, since indexing and searching parts will be unavailable. It can be done using a command “.\elasticsearch.bat” as it is shown on the figure 4.



```
PS C:\ElasticSearch\elasticsearch-7.10.2\bin> .\elasticsearch.bat
```

Figure 4: A screenshot of a process of launching ElasticSearch Engine in Windows PowerShell.

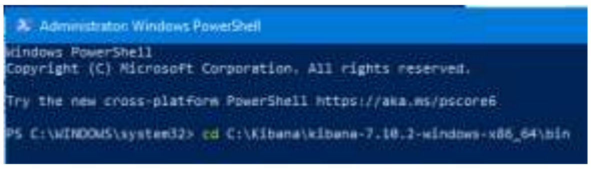
As the result, a user should obtain a following log information regarding the successful launch of the ElasticSearch engine. As it can be seen on the figure 5, to get an access to the server, a user can use an IP address, that is 127.0.0.1:9200.



```
[2021-03-04T12:57:11,332][WARN ][o.e.g.DanglingIndicesState] [DESKTOP-K0D65LT] gateway.auto_import_dangling_indices is disabled, dangling indices will not be automatically detected or imported and must b
[2021-03-04T12:57:14,946][INFO ][o.e.n.Node               ] [DESKTOP-K0D65LT] initialized
[2021-03-04T12:57:14,952][INFO ][o.e.n.Node               ] [DESKTOP-K0D65LT] starting ...
[2021-03-04T12:57:19,271][INFO ][o.e.t.TransportService   ] [DESKTOP-K0D65LT] publish_address {127.0.0.1:9300}, bound_addresses {127.0.0.1:9300}, {[::1]:9300}
[2021-03-04T12:57:23,338][WARN ][o.e.b.BootstrapChecks    ] [DESKTOP-K0D65LT] the default discovery settings are unsuitable for production use; at least one of [discovery.seed_hosts, discovery.seed_provi
 must be configured
[2021-03-04T12:57:23,343][INFO ][o.e.c.c.Coordinator      ] [DESKTOP-K0D65LT] cluster UUID [P8vwA_sW5]-y[illegible]]
[2021-03-04T12:57:23,390][INFO ][o.e.c.c.ClusterBootstrapService] [DESKTOP-K0D65LT] no discovery configuration found, will perform best-effort cluster bootstrapping after [3s] unless existing master is d
[2021-03-04T12:57:24,142][INFO ][o.e.c.s.MasterService    ] [DESKTOP-K0D65LT] elected-as-master ([1] nodes joined)[{DESKTOP-K0D65LT}{mH7E7306Rry-BFApmkO9Rw}{fbajgNTQQYGmGaGB8Cpmww}{127.0.0.1}{127.0.0.1:9
=8274780160, xpack.installed=true, transform.node=true, ml.max_open_jobs=20} elect leader, _BECOME_MASTER_TASK_, _FINISH_ELECTION_], term: 13, version: 1173, delta: master node changed {previous [], curr
-BFApmkO9Rw}{fbajgNTQQYGmGaGB8Cpmww}{127.0.0.1}{127.0.0.1:9300}{cdhilmrstw}{ml.machine_memory=8274780160, xpack.installed=true, transform.node=true, ml.max_open_jobs=20}]}
[2021-03-04T12:57:25,485][INFO ][o.e.c.s.ClusterApplierService] [DESKTOP-K0D65LT] master node changed {previous [], current [{DESKTOP-K0D65LT}{mH7E7306Rry-BFApmkO9Rw}{fbajgNTQQYGmGaGB8Cpmww}{127.0.0.1}{1
ine_memory=8274780160, xpack.installed=true, transform.node=true, ml.max_open_jobs=20}]}, term: 13, version: 1173, reason: Publication{term=13, version=1173}
[2021-03-04T12:57:30,048][INFO ][o.e.l.LicenseService     ] [DESKTOP-K0D65LT] license [aa07bbb0-3a58-4afb-bb3a-9424da2c69a6] mode [basic] - valid
[2021-03-04T12:57:30,065][INFO ][o.e.x.s.s.SecurityStatusChangeListener] [DESKTOP-K0D65LT] Active license is now [BASIC]; Security is disabled
[2021-03-04T12:57:30,108][INFO ][o.e.g.GatewayService     ] [DESKTOP-K0D65LT] recovered [12] indices into cluster_state
[2021-03-04T12:57:31,202][INFO ][o.e.h.AbstractHttpServerTransport] [DESKTOP-K0D65LT] publish_address {127.0.0.1:9200}, bound_addresses {127.0.0.1:9200}, {[::1]:9200}
[2021-03-04T12:57:31,204][INFO ][o.e.n.Node               ] [DESKTOP-K0D65LT] started
[2021-03-04T12:57:38,420][INFO ][o.e.c.r.a.AllocationService] [DESKTOP-K0D65LT] Cluster health status changed from [RED] to [YELLOW] (reason: [shards started [[ilm-history-3-000001][0]]]).
[2021-03-04T12:57:57,393][WARN ][o.e.c.r.a.DiskThresholdMonitor] [DESKTOP-K0D65LT] high disk watermark [90%] exceeded on [mH7E7306Rry-BFApmkO9Rw][DESKTOP-K0D65LT][C:\ElasticSearch\elasticsearch-7.10.2\da
rds will be relocated away from this node; currently relocating away shards totalling [0] bytes; the node is expected to continue to exceed the high disk watermark when these relocations are complete
[2021-03-04T12:59:26,032][WARN ][o.e.c.r.a.DiskThresholdMonitor] [DESKTOP-K0D65LT] high disk watermark [90%] exceeded on [mH7E7306Rry-BFApmkO9Rw][DESKTOP-K0D65LT][C:\ElasticSearch\elasticsearch-7.10.2\da
rds will be relocated away from this node; currently relocating away shards totalling [0] bytes; the node is expected to continue to exceed the high disk watermark when these relocations are complete
[2021-03-04T13:00:56,043][WARN ][o.e.c.r.a.DiskThresholdMonitor] [DESKTOP-K0D65LT] high disk watermark [90%] exceeded on [mH7E7306Rry-BFApmkO9Rw][DESKTOP-K0D65LT][C:\ElasticSearch\elasticsearch-7.10.2\da
rds will be relocated away from this node; currently relocating away shards totalling [0] bytes; the node is expected to continue to exceed the high disk watermark when these relocations are complete
[2021-03-04T13:01:43,848][INFO ][o.e.c.m.MetadataIndexTemplateService] [DESKTOP-K0D65LT] adding template [.management-beats] for index patterns [.management-beats]
[2021-03-04T13:01:56,049][WARN ][o.e.c.r.a.DiskThresholdMonitor] [DESKTOP-K0D65LT] high disk watermark [90%] exceeded on [mH7E7306Rry-BFApmkO9Rw][DESKTOP-K0D65LT][C:\ElasticSearch\elasticsearch-7.10.2\da
rds will be relocated away from this node; currently relocating away shards totalling [0] bytes; the node is expected to continue to exceed the high disk watermark when these relocations are complete
[2021-03-04T13:03:26,058][WARN ][o.e.c.r.a.DiskThresholdMonitor] [DESKTOP-K0D65LT] high disk watermark [90%] exceeded on [mH7E7306Rry-BFApmkO9Rw][DESKTOP-K0D65LT][C:\ElasticSearch\elasticsearch-7.10.2\da
rds will be relocated away from this node; currently relocating away shards totalling [0] bytes; the node is expected to continue to exceed the high disk watermark when these relocations are complete
[2021-03-04T13:04:26,063][WARN ][o.e.c.r.a.DiskThresholdMonitor] [DESKTOP-K0D65LT] high disk watermark [90%] exceeded on [mH7E7306Rry-BFApmkO9Rw][DESKTOP-K0D65LT][C:\ElasticSearch\elasticsearch-7.10.2\da
rds will be relocated away from this node; currently relocating away shards totalling [0] bytes; the node is expected to continue to exceed the high disk watermark when these relocations are complete
```

Figure 5: A screenshot of logs on the sucessfully launched the ElasticSearch engine.

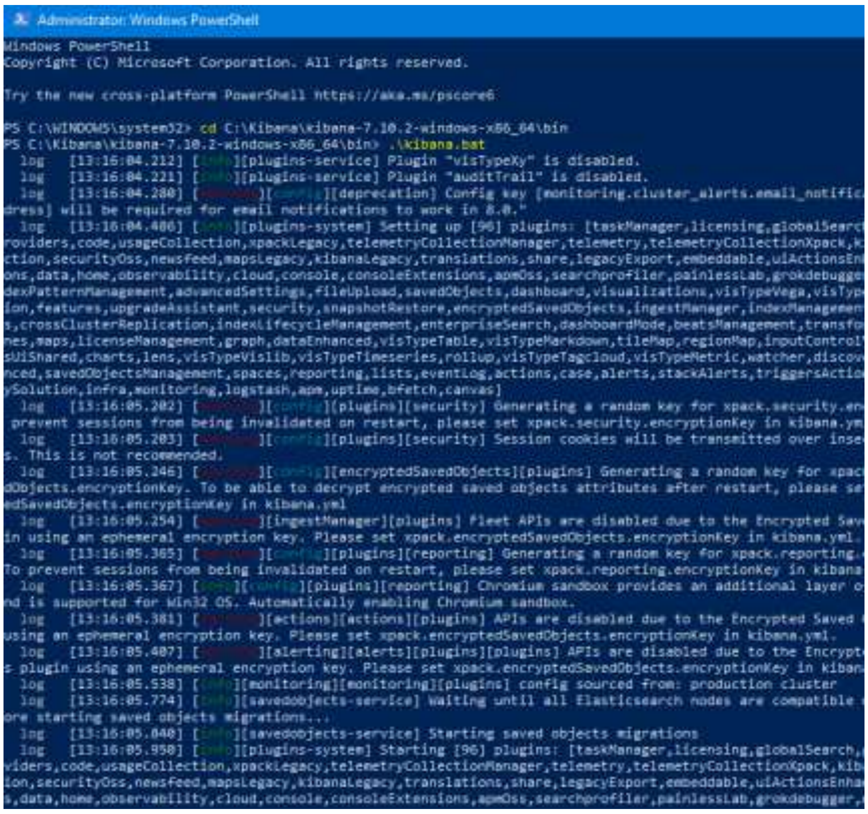
Then, a user has to launch a Kibana to be able to interact with ElasticSearch using a convenient GUI. To run it, he should open a Windows command line and enter the Administrator mode. Having done that, the next step is to enter a directory that contains the Kibana .bat file. It is clearly shown on the figure 6.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> cd C:\Kibana\kibana-7.10.2-windows-x86_64\bin
```

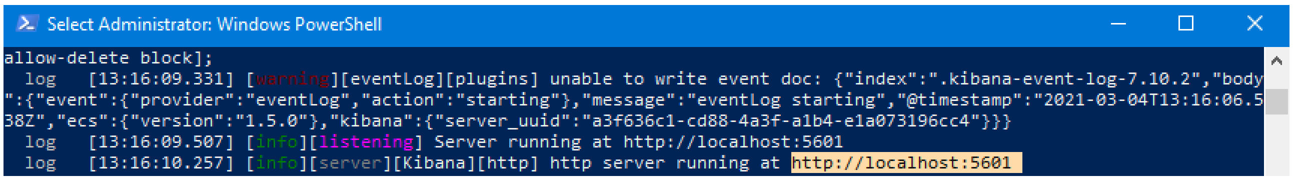
Figure 6: Screenshot of a process of changing directory to access the Kibana .bat file.

Eventually, having entered “.\kibana.bat”, he should obtain following logs and find the address of the Kibana server, as it is shown on figures 7,8.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\WINDOWS\system32> cd C:\Kibana\kibana-7.10.2-windows-x86_64\bin
PS C:\Kibana\kibana-7.10.2-windows-x86_64\bin> .\kibana.bat
  log   [13:16:04.212] [info][plugins-service] Plugin "visTypeXy" is disabled.
  log   [13:16:04.221] [info][plugins-service] Plugin "auditTrail" is disabled.
  log   [13:16:04.280] [warning][config][deprecation] Config key [monitoring.cluster_alerts.email_notific
dress] will be required for email notifications to work in 8.0."
  log   [13:16:04.486] [info][plugins-system] Setting up [96] plugins: [taskManager,licensing,globalSearc
roviders,code,usageCollection,xpackLegacy,telemetryCollectionManager,telemetry,telemetryCollectionXpack,k
ction,securityOss,newsfeed,mapsLegacy,kibanaLegacy,translations,share,legacyExport,embeddable,uiActionsEn
ons,data,home,observability,cloud,console,consoleExtensions,apmOss,searchprofiler,painlessLab,grokdebugge
dexPatternManagement,advancedSettings,fileUpload,savedObjects,dashboard,visualizations,visTypeVega,visTyp
ion,features,upgradeAssistant,security,snapshotRestore,encryptedSavedObjects,ingestManager,indexManagemen
s,crossClusterReplication,indexLifecycleManagement,enterpriseSearch,dashboardMode,beatsManagement,transfo
nes,maps,licenseManagement,graph,dataEnhanced,visTypeTable,visTypeMarkdown,tileMap,regionMap,inputControl
sUiShared,charts,lens,visTypeVislib,visTypeTimeseries,rollup,visTypeTagcloud,visTypeMetric,watcher,discov
nced,savedObjectsManagement,spaces,reporting,lists,eventLog,actions,case,alerts,stackAlerts,triggersActio
ySolution,infra,monitoring,logstash,apm,uptime,bfetch,canvas]
  log   [13:16:05.202] [warning][config][plugins][security] Generating a random key for xpack.security.en
 prevent sessions from being invalidated on restart, please set xpack.security.encryptionKey in kibana.ym
  log   [13:16:05.203] [warning][config][plugins][security] Session cookies will be transmitted over inse
s. This is not recommended.
  log   [13:16:05.246] [warning][config][encryptedSavedObjects][plugins] Generating a random key for xpac
dObjects.encryptionKey. To be able to decrypt encrypted saved objects attributes after restart, please se
edSavedObjects.encryptionKey in kibana.yml
  log   [13:16:05.254] [warning][ingestManager][plugins] Fleet APIs are disabled due to the Encrypted Sav
in using an ephemeral encryption key. Please set xpack.encryptedSavedObjects.encryptionKey in kibana.yml.
  log   [13:16:05.365] [warning][config][plugins][reporting] Generating a random key for xpack.reporting.
To prevent sessions from being invalidated on restart, please set xpack.reporting.encryptionKey in kibana
  log   [13:16:05.367] [info][config][plugins][reporting] Chromium sandbox provides an additional layer o
nd is supported for Win32 OS. Automatically enabling Chromium sandbox.
  log   [13:16:05.381] [warning][actions][actions][plugins] APIs are disabled due to the Encrypted Saved
using an ephemeral encryption key. Please set xpack.encryptedSavedObjects.encryptionKey in kibana.yml.
  log   [13:16:05.407] [warning][alerting][alerts][plugins][plugins] APIs are disabled due to the Encrypt
s plugin using an ephemeral encryption key. Please set xpack.encryptedSavedObjects.encryptionKey in kiban
  log   [13:16:05.538] [info][monitoring][monitoring][plugins] config sourced from: production cluster
  log   [13:16:05.774] [info][savedobjects-service] Waiting until all Elasticsearch nodes are compatible
ore starting saved objects migrations...
  log   [13:16:05.840] [info][savedobjects-service] Starting saved objects migrations
  log   [13:16:05.950] [info][plugins-system] Starting [96] plugins: [taskManager,licensing,globalSearch,
viders,code,usageCollection,xpackLegacy,telemetryCollectionManager,telemetry,telemetryCollectionXpack,kib
ion,securityOss,newsfeed,mapsLegacy,kibanaLegacy,translations,share,legacyExport,embeddable,uiActionsEnha
s,data,home,observability,cloud,console,consoleExtensions,apmOss,searchprofiler,painlessLab,grokdebugger,
```

Figure 7: A screenshot of logs on the sucessfully launched the Kibana server.



```
Select Administrator: Windows PowerShell
allow-delete block];
  log   [13:16:09.331] [warning][eventLog][plugins] unable to write event doc: {"index":".kibana-event-log-7.10.2","body
":{"event":{"provider":"eventLog","action":"starting"},"message":"eventLog starting","@timestamp":"2021-03-04T13:16:06.5
38Z","ecs":{"version":"1.5.0"},"kibana":{"server_uuid":"a3f636c1-cd88-4a3f-a1b4-e1a073196cc4"}}}
  log   [13:16:09.507] [info][listening] Server running at http://localhost:5601
  log   [13:16:10.257] [info][server][Kibana][http] http server running at http://localhost:5601
```

Figure 8: A screenshot of an address of the Kibana server.

Having entered the found web-address, a user should get a following Kibana web-page, as it can be seen on the figure 9.

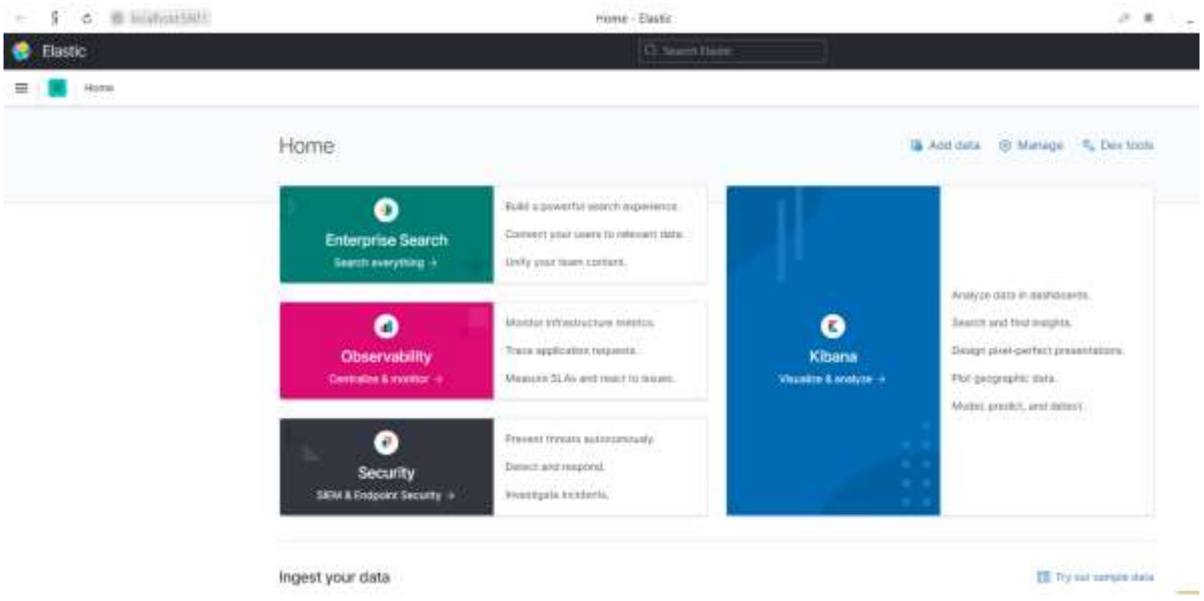

Home - Elastic
Elastic
Search Elastic
Home
Home
Add data
Manage
Dev tools
Enterprise Search
Search everything →
Build a powerful search experience.
Connect your users to relevant data.
Unify your team content.
Observability
Centralize & monitor →
Monitor infrastructure metrics.
Trace application requests.
Measure SLAs and react to issues.
Security
SIEM & Endpoint Security →
Prevent threats autonomously.
Detect and respond.
Investigate incidents.
Kibana
Visualize & analyze →
Analyze data in dashboards.
Search and find insights.
Design pixel-perfect presentations.
Plot geographic data.
Model, predict, and detect.
Ingest your data
Try our sample data

Figure 9: A screenshot of the Kibana GUI.

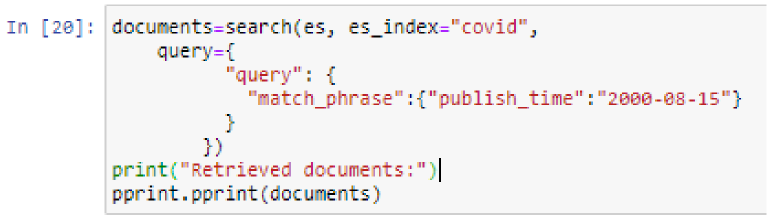
After making these steps and launching the programme-running part of the code, the implemented search system with back-end module – ElasticSearch and GUI – Kibana is ready for performing search queries.

The search can be performed in two ways:

- Searching by the Python programme
- Searching in Kibana

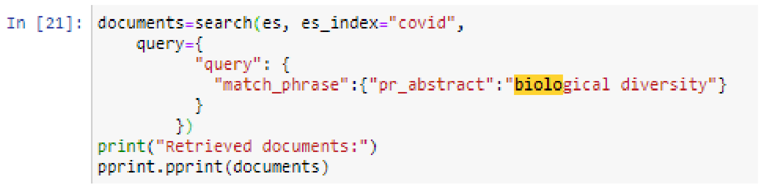
Searching by the Python programme can be performed when a dataset was uploaded and indexed in ElasticSearch. Usually, data are searched in the bottom of the function ”run_program”, but, additionally, when the function “run_program” has done its job, they can be searched by running a code part where the function “search” is called.

A couple of such search examples are depicted on figures. 10,11.



```
In [20]: documents=search(es, es_index="covid",
             query={
                 "query": {
                   "match_phrase":{"publish_time":"2000-08-15"}
                 }
               })
         print("Retrieved documents:")
         pprint.pprint(documents)
```

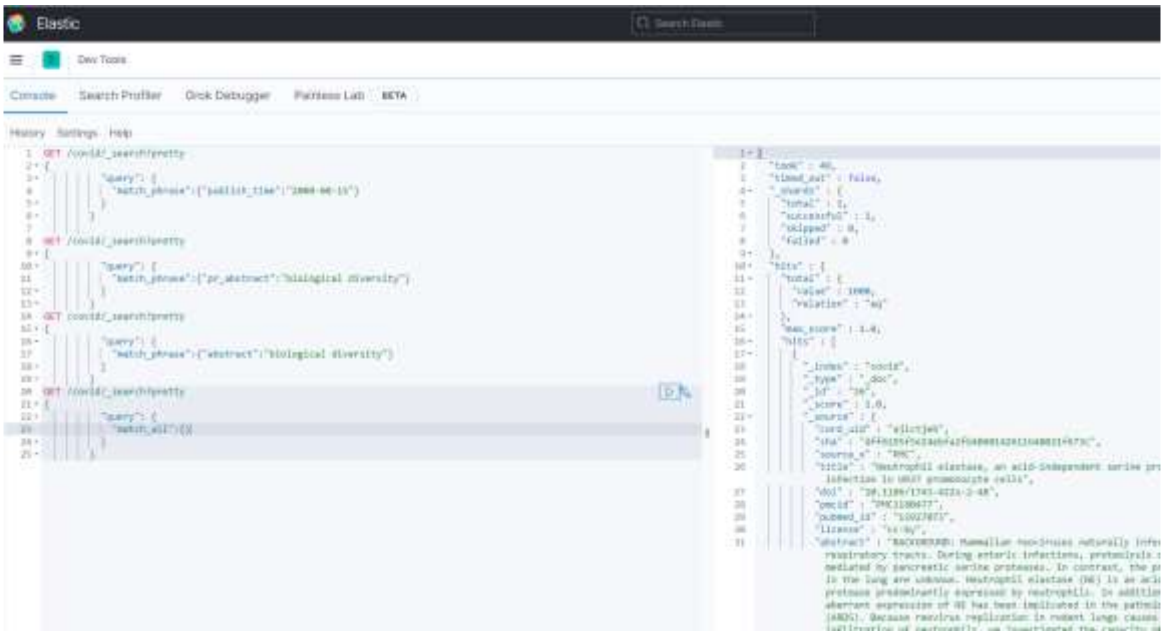
Figure 10: A first example of performing a search by the function “search”.



```
In [21]: documents=search(es, es_index="covid",
             query={
                 "query": {
                   "match_phrase":{"pr_abstract":"biological diversity"}
                 }
               })
         print("Retrieved documents:")
         pprint.pprint(documents)
```

Figure 11: A second example of performing a search by the function “search”.

Searching in Kibana GUI, as it is shown on the figure 12, can be performed when a dataset was uploaded and indexed in ElasticSearch.



```
GET /covid/_search?pretty
{
        "query": {
          "match_phrase":{"publish_time":"2000-08-15"}
        }
      }

GET /covid/_search?pretty
{
        "query": {
          "match_phrase":{"pr_abstract":"biological diversity"}
        }
      }
GET covid/_search?pretty
{
        "query": {
          "match_phrase":{"abstract":"biological diversity"}
        }
      }
GET /covid/_search?pretty
{
        "query": {
          "match_all":{}
        }
      }
```

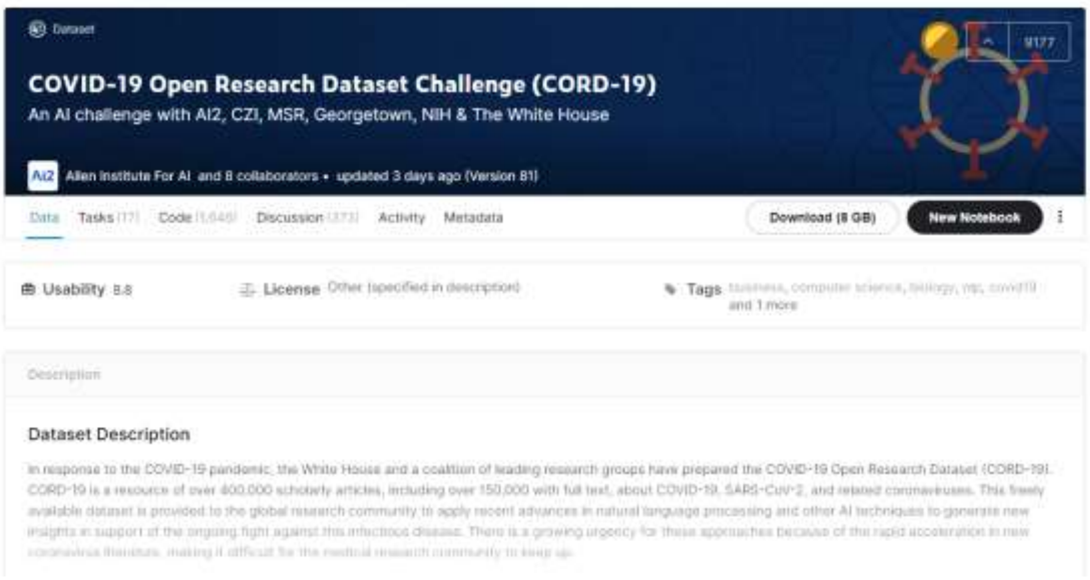
```
{
  "took" : 40,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 1000,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "covid",
        "_type" : "_doc",
        "_id" : "30",
        "_score" : 1.0,
        "_source" : {
          "cord_uid" : "[illegible]",
          "sha" : "[illegible]",
          "source_x" : "PMC",
          "title" : "Neutrophil elastase, an acid-independent serine pr
            infection in [illegible] cells",
          "doi" : "[illegible]",
          "pmcid" : "[illegible]",
          "pubmed_id" : "[illegible]",
          "license" : "cc-by",
          "abstract" : "BACKGROUND: Mammalian [illegible] naturally infe
            respiratory tracts. During enteric infections, proteolysis
            mediated by pancreatic serine proteases. In contrast, the p
            in the lung are unknown. Neutrophil elastase (NE) is an aci
            protease predominantly expressed by neutrophils. In additio
            aberrant expression of NE has been implicated in the pathol
            (ARDS). Because reovirus replication in rodent lungs causes
            infiltration of neutrophils, we investigated the capacity o
```

Figure 12: Running search examples in the Kibana GUI.

Indexing

Include here the details of how you download your dataset and index it including any issue that you had and how did you face it. Explain which documents have you selected for your experiments.. You may include screenshots to clarify.

To work on the assignment 1 there was previously downloaded a CORD-19 dataset that contains thousands of scientific articles dedicated to a research of methods for fight against the COVID-19 disease. Since, the archive contains a massive amount of articles, it was rather hard and took much of time to download and extract it. On figures 13,14 below there is a shown description of this corpus, its structure and a volume of data.



Dataset

COVID-19 Open Research Dataset Challenge (CORD-19)

An AI challenge with AI2, CZI, MSR, Georgetown, NIH & The White House

AI2 Allen Institute For AI and 8 collaborators • updated 3 days ago (Version 81)

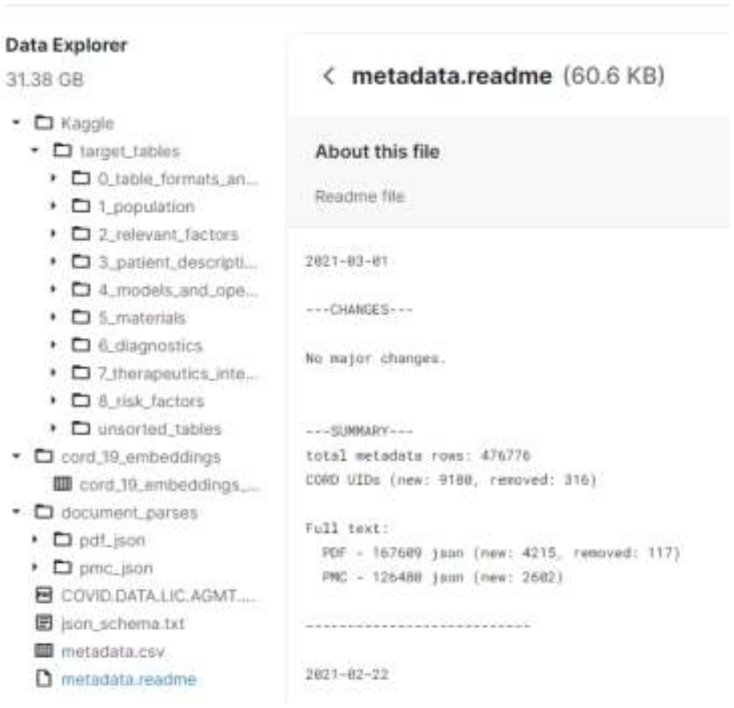
Usability 8.8 | License Other (specified in description) | Tags business, computer science, biology, nlp, covid19 and 1 more

Description

Dataset Description

In response to the COVID-19 pandemic, the White House and a coalition of leading research groups have prepared the COVID-19 Open Research Dataset (CORD-19). CORD-19 is a resource of over 400,000 scholarly articles, including over 150,000 with full text, about COVID-19, SARS-CoV-2, and related coronaviruses. This freely available dataset is provided to the global research community to apply recent advances in natural language processing and other AI techniques to generate new insights in support of the ongoing fight against this infectious disease. There is a growing urgency for these approaches because of the rapid acceleration in new coronavirus literature, making it difficult for the medical research community to keep up.

Figure 13: A description of the CORD-19 purpose from its source on the Kaggle website.



Data Explorer

31.38 GB

```
▾ Kaggle
  ▾ target_tables
    ▸ 0_table_formats_an...
    ▸ 1_population
    ▸ 2_relevant_factors
    ▸ 3_patient_descripti...
    ▸ 4_models_and_ope...
    ▸ 5_materials
    ▸ 6_diagnostics
    ▸ 7_therapeutics_inte...
    ▸ 8_risk_factors
    ▸ unsorted_tables
▾ cord_19_embeddings
    cord_19_embeddings_...
▾ document_parses
  ▸ pdf_json
  ▸ pmc_json
  COVID.DATA.LIC.AGMT....
  json_schema.txt
  metadata.csv
  metadata.readme
```

metadata.readme (60.6 KB)

About this file

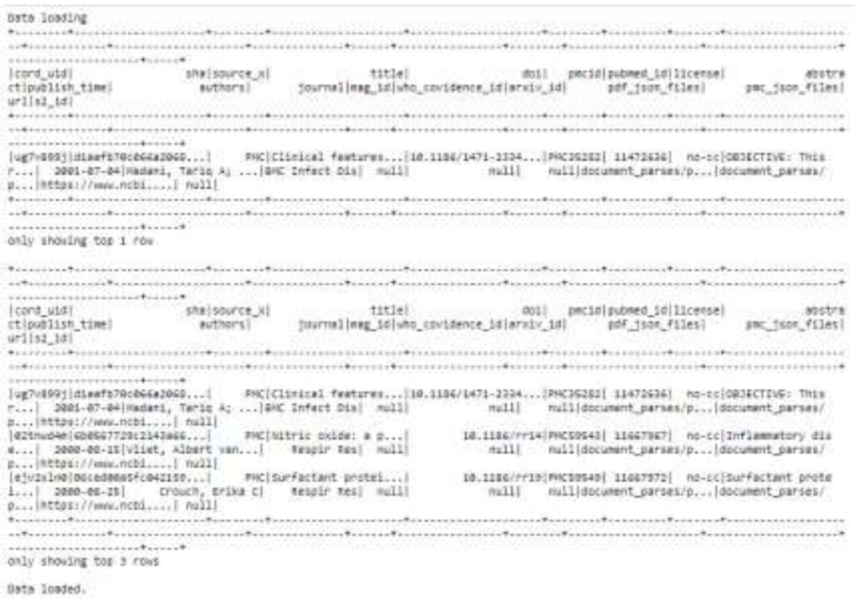
Readme file

```
2021-03-01

---CHANGES---

No major changes.


---SUMMARY---
total metadata rows: 476776
CORD UIDs (new: 9180, removed: 316)

Full text:
  PDF - 167609 json (new: 4215, removed: 117)
  PMC - 126480 json (new: 2602)

--------------------------

2021-02-22
```

Figure 14: A description of the CORD-19 structure on the Kaggle website.

Having obtained all required information, there was loaded a sample of first 1000 articles from the metadata.csv file using a pyspark. This is a Python library that allows developers to load and structure data into dataframes providing a high level of efficiency and performance.

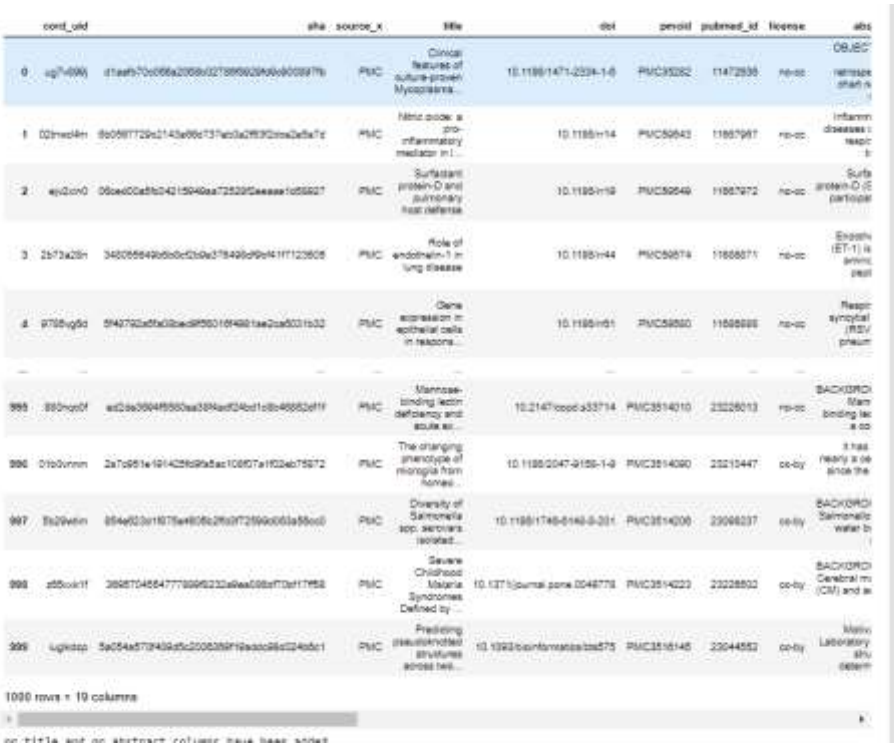
In the result, the obtained spreadsheet has a following view that can be seen on the figure 15.



```
Data loading
+--------+--------------------+--------+--------------------+--------------------+--------+---------+-------+--------------------
--+------------+--------------------+--------------+------+--------------------+--------+--------------------+--------------------+
--------------------+-----+
|cord_uid|                 sha|source_x|               title|                 doi|   pmcid|pubmed_id|license|              abstra
ct|publish_time|             authors|       journal|mag_id|who_covidence_id|arxiv_id|      pdf_json_files|      pmc_json_files|
url|s2_id|
+--------+--------------------+--------+--------------------+--------------------+--------+---------+-------+--------------------
--+------------+--------------------+--------------+------+--------------------+--------+--------------------+--------------------+
--------------------+-----+
|ug7v899j|d1aafb70c066a2068...|     PMC|Clinical features...|10.1186/1471-2334...|PMC35282| 11472636|  no-cc|OBJECTIVE: This
r...|  2001-07-04|Madani, Tariq A; ...|BMC Infect Dis|  null|                null|    null|document_parses/p...|document_parses/
p...|https://www.ncbi....| null|
+--------+--------------------+--------+--------------------+--------------------+--------+---------+-------+--------------------
--+------------+--------------------+--------------+------+--------------------+--------+--------------------+--------------------+
--------------------+-----+
only showing top 1 row

+--------+--------------------+--------+--------------------+--------------------+--------+---------+-------+--------------------
--+------------+--------------------+--------------+------+--------------------+--------+--------------------+--------------------+
--------------------+-----+
|cord_uid|                 sha|source_x|               title|                 doi|   pmcid|pubmed_id|license|              abstra
ct|publish_time|             authors|       journal|mag_id|who_covidence_id|arxiv_id|      pdf_json_files|      pmc_json_files|
url|s2_id|
+--------+--------------------+--------+--------------------+--------------------+--------+---------+-------+--------------------
--+------------+--------------------+--------------+------+--------------------+--------+--------------------+--------------------+
--------------------+-----+
|ug7v899j|d1aafb70c066a2068...|     PMC|Clinical features...|10.1186/1471-2334...|PMC35282| 11472636|  no-cc|OBJECTIVE: This
r...|  2001-07-04|Madani, Tariq A; ...|BMC Infect Dis|  null|                null|    null|document_parses/p...|document_parses/
p...|https://www.ncbi....| null|
|02tnwd4m|6b0567729c2143a66...|     PMC|Nitric oxide: a p...|        10.1186/rr14|PMC59543| 11667967|  no-cc|Inflammatory dis
e...|  2000-08-15|Vliet, Albert van...|     Respir Res|  null|                null|    null|document_parses/p...|document_parses/
p...|https://www.ncbi....| null|
|ejv2xln0|06ced00a5fc042150...|     PMC|Surfactant protei...|        10.1186/rr19|PMC59549| 11667972|  no-cc|Surfactant prote
i...|  2000-08-25|    Crouch, Erika C|     Respir Res|  null|                null|    null|document_parses/p...|document_parses/
p...|https://www.ncbi....| null|
+--------+--------------------+--------+--------------------+--------------------+--------+---------+-------+--------------------
--+------------+--------------------+--------------+------+--------------------+--------+--------------------+--------------------+
--------------------+-----+
only showing top 3 rows

Data loaded.
```

Figure 15: A raw view of first 3 rows from the metadata.csv table.

As it is depicted on the figure above, data represented there have a rather corrupted view. Therefore, there was made a decision to fix this issue by converting it to a Pandas dataframe. This resulted into the much clear representation on the figure 16 below.

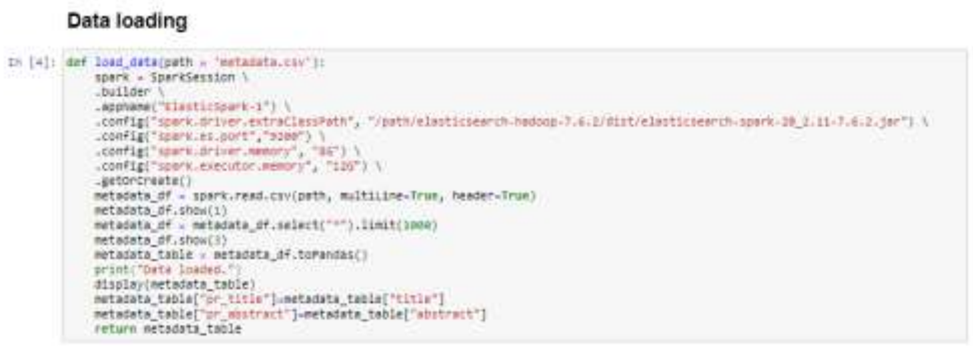


	cord_uid	sha	source_x	title	doi	pmcid	pubmed_id	license	abs
0	ug7v899j	d1aafb70c066a2068b02786f8929fd9c900897fb	PMC	Clinical features of culture-proven Mycoplasma...	10.1186/1471-2334-1-6	PMC35282	11472636	no-cc	OBJEC retrospe chart r
1	02tnwd4m	6b0567729c2143a66d737eb0a2f63f2dce2e5a7d	PMC	Nitric oxide: a pro-inflammatory mediator in l...	10.1186/rr14	PMC59543	11667967	no-cc	Inflamm diseases respir
2	ejv2xln0	06ced00a5fc04215949aa72528f2eeaae1d58927	PMC	Surfactant protein-D and pulmonary host defense	10.1186/rr19	PMC59549	11667972	no-cc	Surfa protein-D (S participat
3	2b73a28n	348055649b6b8cf2b9a376498df9bf41f7123605	PMC	Role of endothelin-1 in lung disease	10.1186/rr44	PMC59574	11686871	no-cc	Endoth (ET-1) is amino pept
4	9785vg6d	5f48792a5fa08bed9f56016f4981ae2ca6031b32	PMC	Gene expression in epithelial cells in respons...	10.1186/rr61	PMC59580	11686888	no-cc	Respir syncytial (RSV pneum
...	...	...	...	...	...	...	...	...	...
995	[illegible]	[illegible]	PMC	Mannose-binding lectin deficiency and acute ex...	10.2147/copd.s33714	PMC3514010	23226013	no-cc	BACKGROU Man binding lec a co
996	01b0vnmm	[illegible]	PMC	The changing phenotype of microglia from homeo...	10.1186/2047-9158-1-9	PMC3514090	23210447	cc-by	It has nearly a de since the
997	[illegible]	[illegible]	PMC	Diversity of Salmonella spp. serovars isolated...	10.1186/1746-6148-8-201	PMC3514206	23098237	cc-by	BACKGROU Salmonella water b
998	[illegible]	[illegible]	PMC	Severe Childhood Malaria Syndromes Defined by ...	10.1371/journal.pone.0048778	PMC3514223	23226802	cc-by	BACKGROU Cerebral m (CM) and a
999	[illegible]	[illegible]	PMC	Predicting pseudoknotted structures across two...	10.1093/bioinformatics/bts575	PMC3516146	23044552	cc-by	Motiv Laboratory stru determ

1000 rows × 19 columns

Figure 16: A smooth view of the metadata.csv table content.

On the figure 17 below a content of the function “load_data” is clearly represented.



Data loading

```
In [4]: def load_data(path = 'metadata.csv'):
            spark = SparkSession \
            .builder \
            .appName("ElasticSpark-1") \
            .config("spark.driver.extraClassPath", "/path/elasticsearch-hadoop-7.6.2/dist/elasticsearch-spark-20_2.11-7.6.2.jar") \
            .config("spark.es.port","9200") \
            .config("spark.driver.memory", "8G") \
            .config("spark.executor.memory", "12G") \
            .getOrCreate()
            metadata_df = spark.read.csv(path, multiLine=True, header=True)
            metadata_df.show(1)
            metadata_df = metadata_df.select("*").limit(1000)
            metadata_df.show(3)
            metadata_table = metadata_df.toPandas()
            print("Data loaded.")
            display(metadata_table)
            metadata_table["pr_title"]=metadata_table["title"]
            metadata_table["pr_abstract"]=metadata_table["abstract"]
            return metadata_table
```

Figure 17: A smooth view of the metadata.csv table content.

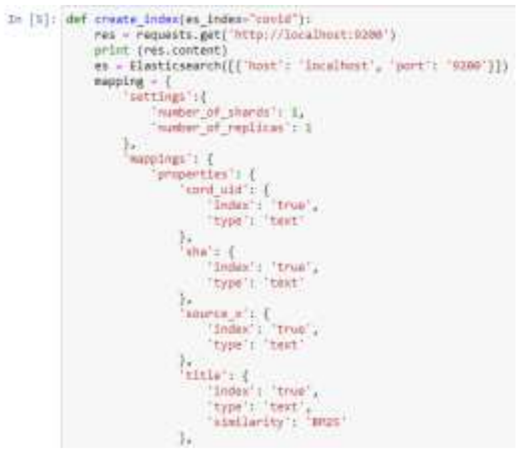
To index the data properly there was made a decision to create mappings for each column of the obtained table. These mappings point out what is a type of a certain variable and whether it should be indexed or no.

Also, it is of importance to specify a similarity function for columns that contain preferably text data. The metadata.csv table has two of such columns:

- title
- abstract

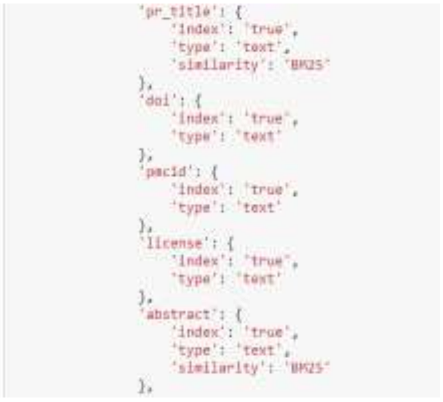
Therefore, it is required to assign these fields a "BM-25" similarity function that performs slightly better than "classic" that is also known as TF/IDF similarity function.

On figures 18,19,20,21 there were displayed four parts of the structure of aforementioned mappings for data indexing.



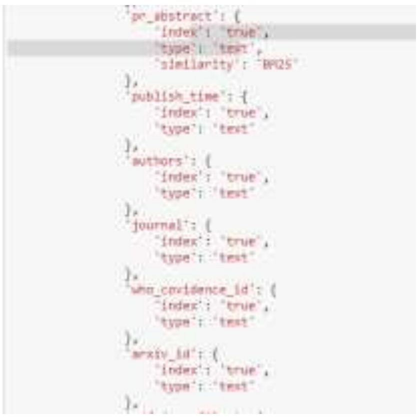
```
In [5]: def create_index(es_index="covid"):
            res = requests.get('http://localhost:9200')
            print (res.content)
            es = Elasticsearch([{'host': 'localhost', 'port': '9200'}])
            mapping = {
                'settings':{
                    'number_of_shards': 1,
                    'number_of_replicas': 1
                },
                'mappings': {
                    'properties': {
                        'cord_uid': {
                            'index': 'true',
                            'type': 'text'
                        },
                        'sha': {
                            'index': 'true',
                            'type': 'text'
                        },
                        'source_x': {
                            'index': 'true',
                            'type': 'text'
                        },
                        'title': {
                            'index': 'true',
                            'type': 'text',
                            'similarity': 'BM25'
                        },
```

Figure 18: The mappings structure – the first part.



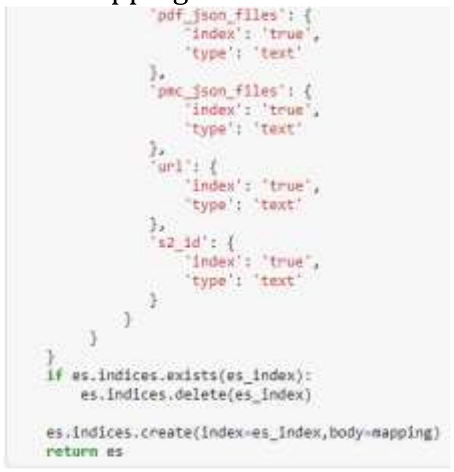
```
                        'pr_title': {
                            'index': 'true',
                            'type': 'text',
                            'similarity': 'BM25'
                        },
                        'doi': {
                            'index': 'true',
                            'type': 'text'
                        },
                        'pmcid': {
                            'index': 'true',
                            'type': 'text'
                        },
                        'license': {
                            'index': 'true',
                            'type': 'text'
                        },
                        'abstract': {
                            'index': 'true',
                            'type': 'text',
                            'similarity': 'BM25'
                        },
```

Figure 19: The mappings structure – the second part.



```
        'pr_abstract': {
            'index': 'true',
            'type': 'text',
            'similarity': 'BM25'
        },
        'publish_time': {
            'index': 'true',
            'type': 'text'
        },
        'authors': {
            'index': 'true',
            'type': 'text'
        },
        'journal': {
            'index': 'true',
            'type': 'text'
        },
        'who_covidence_id': {
            'index': 'true',
            'type': 'text'
        },
        'arxiv_id': {
            'index': 'true',
            'type': 'text'
        },
```

Figure 20: The mappings structure – the third part.



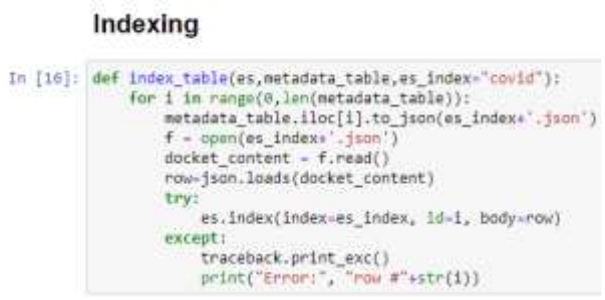
```
            'pdf_json_files': {
                'index': 'true',
                'type': 'text'
            },
            'pmc_json_files': {
                'index': 'true',
                'type': 'text'
            },
            'url': {
                'index': 'true',
                'type': 'text'
            },
            's2_id': {
                'index': 'true',
                'type': 'text'
            }
        }
    }
}
if es.indices.exists(es_index):
    es.indices.delete(es_index)

es.indices.create(index=es_index,body=mapping)
return es
```

Figure 21: The mappings structure – the fourth part.

As it could be noticed, the mappings structure has two extra fields:

- "pr_title"
- "pr_abstract"

These two fields are copies of original ones and are to be processed to present a difference between search results on two versions of a text.
Eventually, when index was created based on those mappings, the data can be uploaded to ElasticSearch using a function that is represented on the figure 22 below.

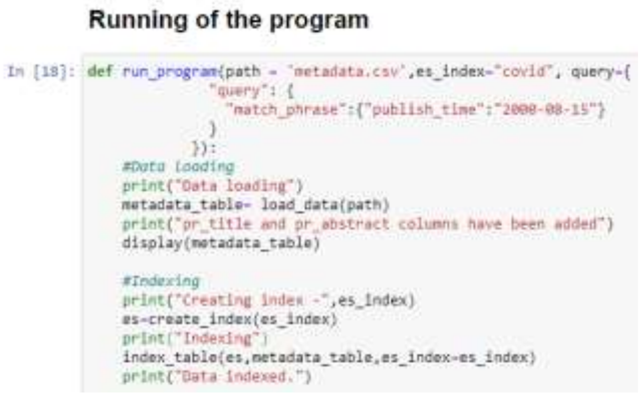


Indexing

```
In [16]: def index_table(es,metadata_table,es_index="covid"):
             for i in range(0,len(metadata_table)):
                 metadata_table.iloc[i].to_json(es_index+'.json')
                 f = open(es_index+'.json')
                 docket_content = f.read()
                 row=json.loads(docket_content)
                 try:
                     es.index(index=es_index, id=i, body=row)
                 except:
                     traceback.print_exc()
                     print("Error:", "row #"+str(i))
```

Figure 22: A function to upload data to ElasticSearch and index them.

On the figure 23 below there is provided a part of the code that is responsible for making "Data loading" and "Indexing" steps. Therefore, a user can decide how to work with these parts.



Running of the program

```
In [18]: def run_program(path = 'metadata.csv',es_index="covid", query={
                    "query": {
                      "match_phrase":{"publish_time":"2000-08-15"}
                    }
                  }):
             #Data Loading
             print("Data loading")
             metadata_table= load_data(path)
             print("pr_title and pr_abstract columns have been added")
             display(metadata_table)

             #Indexing
             print("Creating index -",es_index)
             es=create_index(es_index)
             print("Indexing")
             index_table(es,metadata_table,es_index=es_index)
             print("Data indexed.")
```

Figure 23: A part of the code to perform "Data loading" and "Indexing" steps

Sentence Splitting, Tokenization and Normalization

Sentence splitting went well because it managed to split sentences in the text, providing more efficient keywords selection further

Sentence splitting also went bad because for some sentences where abbreviations occur it split those abbreviations. In general, this problem may not be too critical because there are still the other words to search apart from these ones. For the further programme development, it is worthy finding a certain smart algorithm or a semantic model that could help to address this issue.

Tokenisation went well because by gathering all words occurring in a one sentence it allowed to perform other pre-processing methods such as Lemmatisation or keywords selection, and made it possible to transform the text into a desired form.

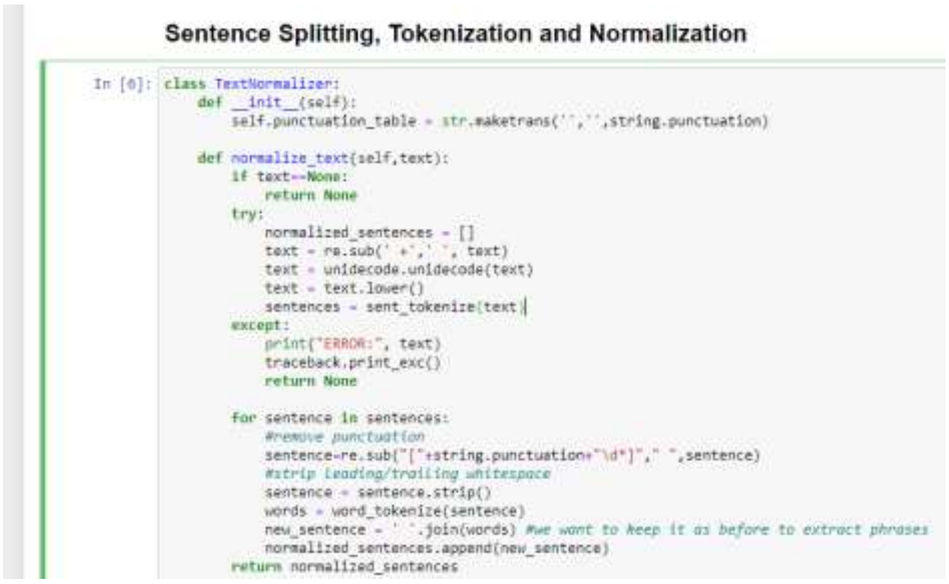
Despite this, tokenisation went bad, because it was rather hard to establish right links between two or even several words in the "Selecting keywords" part. Therefore, in the next chapter there will be more detailed explanation how this problem was particularly solved.

Normalisation went fine because it all words in texts became lowercased. As the result, this technique substantially simplified further text pre-processing and keywords selection steps.

On the other hand, a majority of words with a first capital letter merely lost their meaning due to such a rough simplification. Although, this problem was also particularly solved in the next chapter.

On the figure 24, there was depicted a structure for the "TextNormalizer" class. His responsibilities include removing punctuation signs, casting a text to a lowercase, dividing a text into several sentences and words tokenisation. And, on figures 25,26, this class is used.

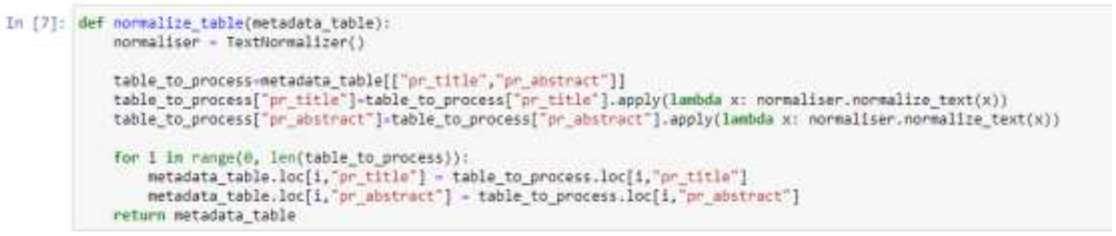
In the result, on the figure 27 a comparative table of differences between normalised text data and not normalised text data is depicted.



Sentence Splitting, Tokenization and Normalization

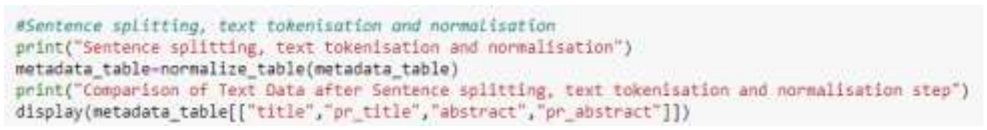
```
In [6]: class TextNormalizer:
            def __init__(self):
                self.punctuation_table = str.maketrans('','',string.punctuation)

            def normalize_text(self,text):
                if text==None:
                    return None
                try:
                    normalized_sentences = []
                    text = re.sub(' +',' ', text)
                    text = unidecode.unidecode(text)
                    text = text.lower()
                    sentences = sent_tokenize(text)
                except:
                    print("ERROR:", text)
                    traceback.print_exc()
                    return None

                for sentence in sentences:
                    #remove punctuation
                    sentence=re.sub("["+string.punctuation+"\d*]"," ",sentence)
                    #strip leading/trailing whitespace
                    sentence = sentence.strip()
                    words = word_tokenize(sentence)
                    new_sentence = ' '.join(words) #we want to keep it as before to extract phrases
                    normalized_sentences.append(new_sentence)
                return normalized_sentences
```

Figure 24: A description of the "TextNormalizer" class



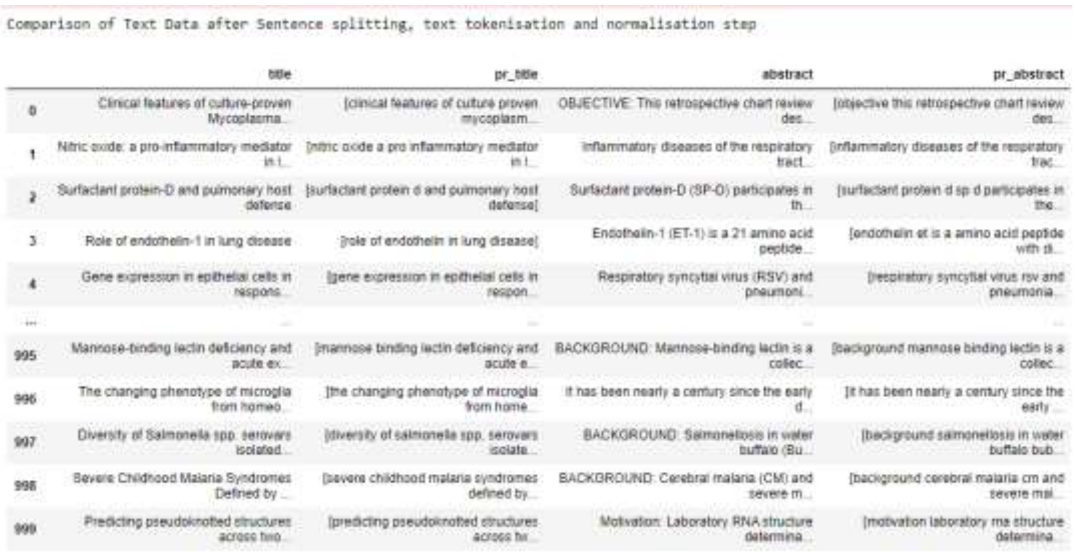
```
In [7]: def normalize_table(metadata_table):
            normaliser = TextNormalizer()

            table_to_process=metadata_table[["pr_title","pr_abstract"]]
            table_to_process["pr_title"]=table_to_process["pr_title"].apply(lambda x: normaliser.normalize_text(x))
            table_to_process["pr_abstract"]=table_to_process["pr_abstract"].apply(lambda x: normaliser.normalize_text(x))

            for i in range(0, len(table_to_process)):
                metadata_table.loc[i,"pr_title"] = table_to_process.loc[i,"pr_title"]
                metadata_table.loc[i,"pr_abstract"] = table_to_process.loc[i,"pr_abstract"]
            return metadata_table
```

Figure 25: A part of the code to call the "TextNormalizer" class



```
#Sentence splitting, text tokenisation and normalisation
print("Sentence splitting, text tokenisation and normalisation")
metadata_table=normalize_table(metadata_table)
print("Comparison of Text Data after Sentence splitting, text tokenisation and normalisation step")
display(metadata_table[["title","pr_title","abstract","pr_abstract"]])
```

Figure 26: A part of the code to perform the normalisation step



```
Comparison of Text Data after Sentence splitting, text tokenisation and normalisation step
```

	title	pr_title	abstract	pr_abstract
0	Clinical features of culture-proven Mycoplasma…	[clinical features of culture proven mycoplasm…	OBJECTIVE: This retrospective chart review des…	[objective this retrospective chart review des…
1	Nitric oxide: a pro-inflammatory mediator in l…	[nitric oxide a pro inflammatory mediator in l…	Inflammatory diseases of the respiratory tract…	[inflammatory diseases of the respiratory trac…
2	Surfactant protein-D and pulmonary host defense	[surfactant protein d and pulmonary host defense]	Surfactant protein-D (SP-D) participates in th…	[surfactant protein d sp d participates in the…
3	Role of endothelin-1 in lung disease	[role of endothelin in lung disease]	Endothelin-1 (ET-1) is a 21 amino acid peptide…	[endothelin et is a amino acid peptide with di…
4	Gene expression in epithelial cells in respons…	[gene expression in epithelial cells in respon…	Respiratory syncytial virus (RSV) and pneumoni…	[respiratory syncytial virus rsv and pneumonia…
…	…	…	…	…
995	Mannose-binding lectin deficiency and acute ex…	[mannose binding lectin deficiency and acute e…	BACKGROUND: Mannose-binding lectin is a collec…	[background mannose binding lectin is a collec…
996	The changing phenotype of microglia from homeo…	[the changing phenotype of microglia from home…	It has been nearly a century since the early d…	[it has been nearly a century since the early …
997	Diversity of Salmonella spp. serovars isolated…	[diversity of salmonella spp. serovars isolate…	BACKGROUND: Salmonellosis in water buffalo (Bu…	[background salmonellosis in water buffalo bub…
998	Severe Childhood Malaria Syndromes Defined by …	[severe childhood malaria syndromes defined by…	BACKGROUND: Cerebral malaria (CM) and severe m…	[background cerebral malaria cm and severe mal…
999	Predicting pseudoknotted structures across two…	[predicting pseudoknotted structures across tw…	Motivation: Laboratory RNA structure determina…	[motivation laboratory rna structure determina…

Figure 27: A comparative table of differences between normalised text data and not normalised text data

Selecting Keywords

Selecting keywords went well because due to a certain bunch of different implemented keywords selection techniques it managed to preserve some sense characterising an article, and made it much simpler for a search engine to find a relevant document.

On the other hand, it went wrong while using TF/IDF because, a TF/IDF method considers frequency of single words – not phrases. However, this problem was solved by replacing whitespaces between words of phrases with a "_" sign. Thus, this trick allowed to address that issue and optimise TF/IDF for that kind of words.

Concerning a solution to problems mentioned in the previous chapter, an implemented keywords selection technique – Text Rank made it possible in a certain extent to preserve links between two or even more words. Due to its unusual approach of designing of semantic graphs it allowed to retain a very sense of a considered sentence.

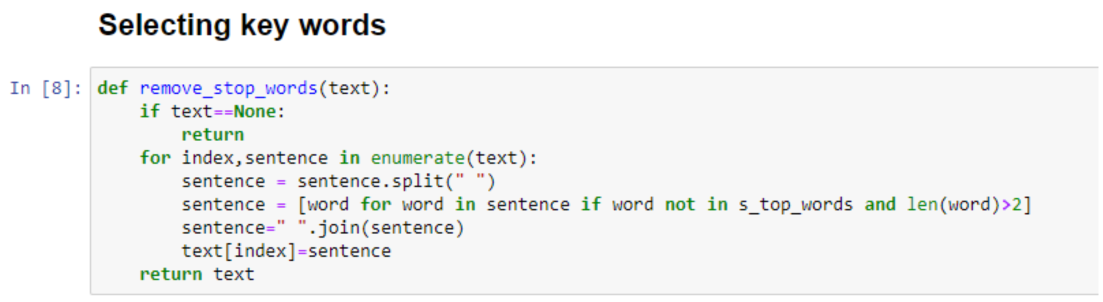
Since this technique evaluates a rank for each word or phrase in a sentence, it was rather convenient to filter those words that make up a noise and retain the most important ones.

Another approach calculates a pointwise mutual information for each bigram or trigram and provides a more efficient keywords selection in comparison with a straightforward bigrams or trigrams frequency distribution, since it considers more important those infrequent words that make sense when they stand next to each other.

In this project, both these methods were implemented and combined to provide much more efficiency for keywords sampling. The idea was to use pros of the Text Rank method to gather only meaningful words or large phrases and then to supplement obtained sentences with keywords gathered by PMI method. Thus, texts should preserve its sense and it would have all important keywords to search.

In addition, a TF/IDF method was also applied to those texts to make them much simpler to search and remove the rest of unimportant words.

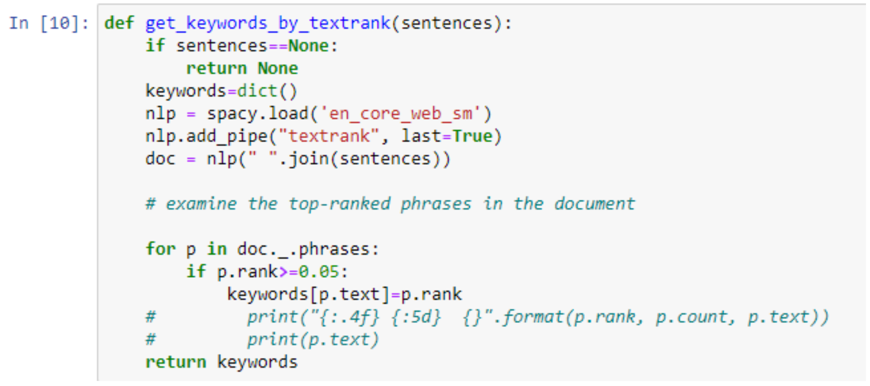
On the figure 28 a function, that tokenise sentences in order to remove any stop words, is displayed.


Selecting key words

```
In [8]: def remove_stop_words(text):
            if text==None:
                return
            for index,sentence in enumerate(text):
                sentence = sentence.split(" ")
                sentence = [word for word in sentence if word not in s_top_words and len(word)>2]
                sentence=" ".join(sentence)
                text[index]=sentence
            return text
```

Figure 28: A function that removes stop words.

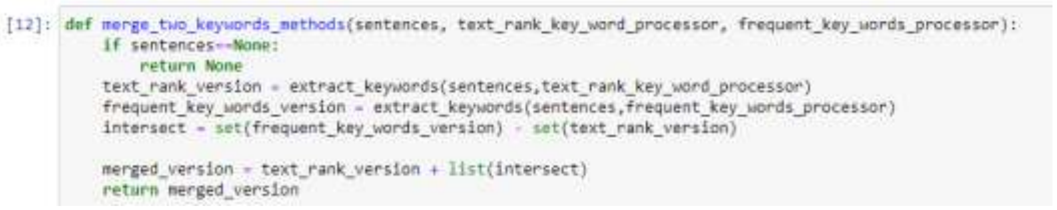
A function, where the TextRank keywords extraction method was implemented, can be seen on the figure 29. From this, we can notice that TextRank uses English language model to do perform its task.



```
In [10]: def get_keywords_by_textrank(sentences):
             if sentences==None:
                 return None
             keywords=dict()
             nlp = spacy.load('en_core_web_sm')
             nlp.add_pipe("textrank", last=True)
             doc = nlp(" ".join(sentences))

             # examine the top-ranked phrases in the document

             for p in doc._.phrases:
                 if p.rank>=0.05:
                     keywords[p.text]=p.rank
             #         print("{:.4f} {:5d}  {}".format(p.rank, p.count, p.text))
             #         print(p.text)
             return keywords
```

Figure 29: A function of keywords extraction by TextRank.

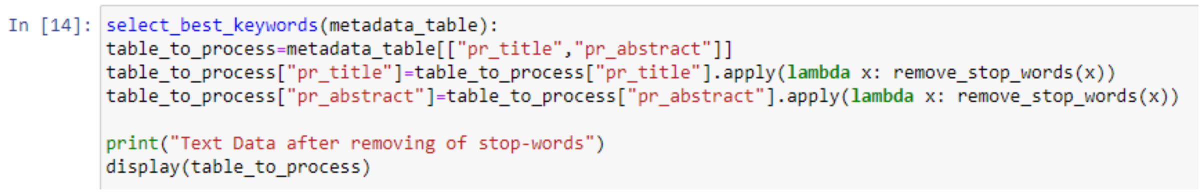
On the figure 30, it can be seen how results of two methods of keywords extraction merge.



```
[12]: def merge_two_keywords_methods(sentences, text_rank_key_word_processor, frequent_key_words_processor):
          if sentences==None:
              return None
          text_rank_version = extract_keywords(sentences,text_rank_key_word_processor)
          frequent_key_words_version = extract_keywords(sentences,frequent_key_words_processor)
          intersect = set(frequent_key_words_version) - set(text_rank_version)

          merged_version = text_rank_version + list(intersect)
          return merged_version
```

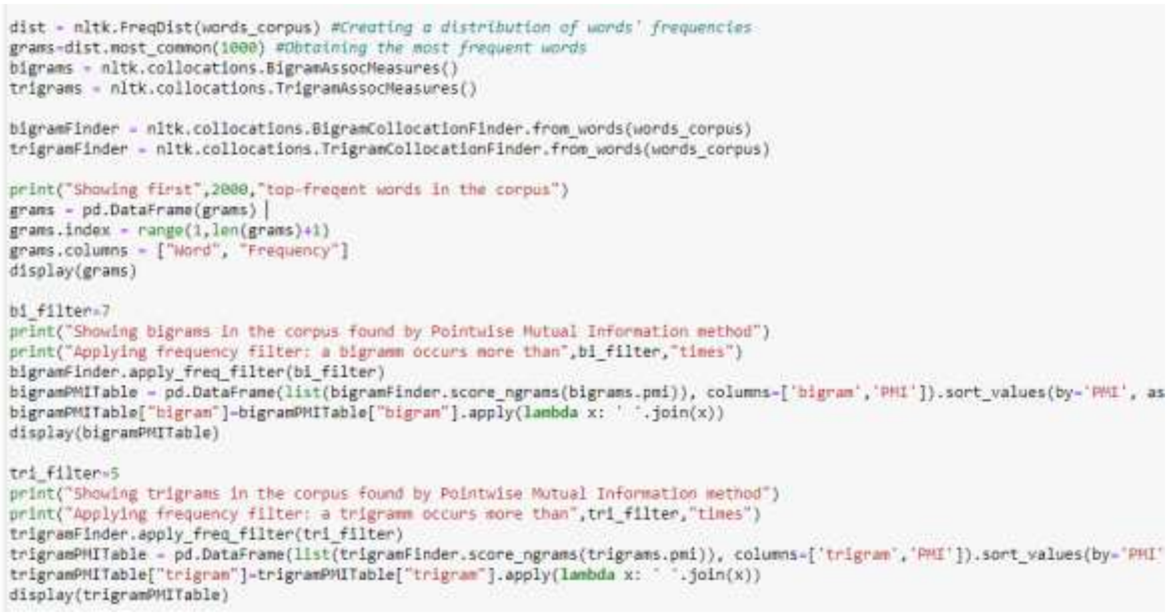
Figure 30: A function that merges results of the PMI method and the TextRank one.

On figures 31,32,33 and 34 we can see the main parts of the function that does a “keywords selection” step.



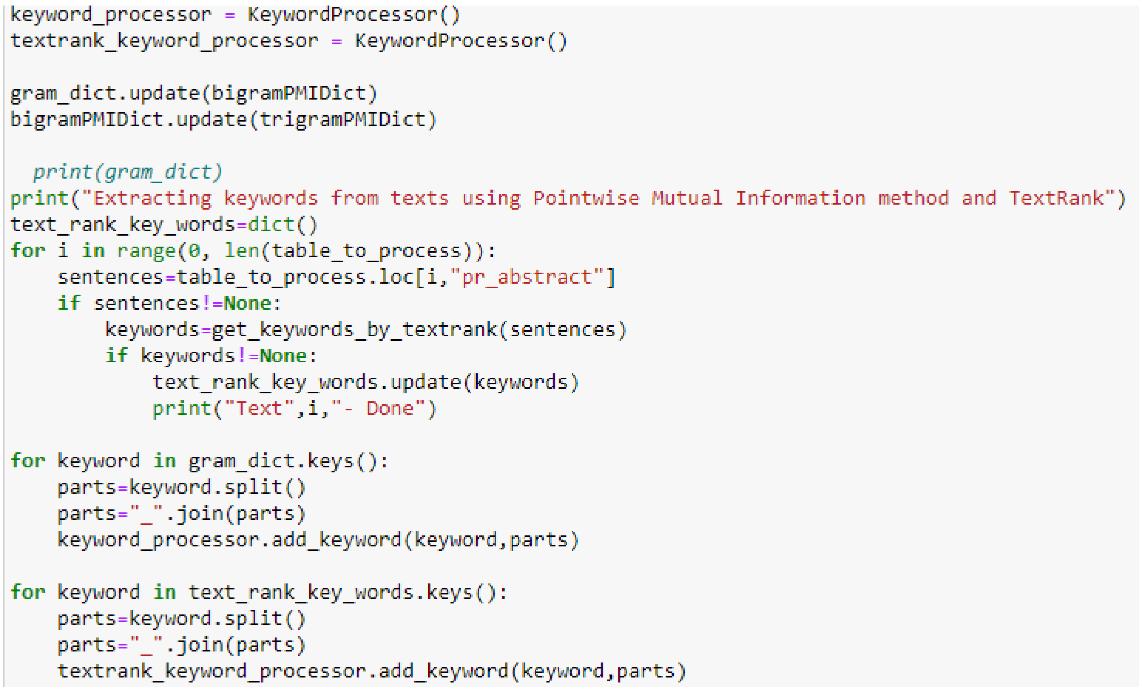
```
In [14]: select_best_keywords(metadata_table):
         table_to_process=metadata_table[["pr_title","pr_abstract"]]
         table_to_process["pr_title"]=table_to_process["pr_title"].apply(lambda x: remove_stop_words(x))
         table_to_process["pr_abstract"]=table_to_process["pr_abstract"].apply(lambda x: remove_stop_words(x))

         print("Text Data after removing of stop-words")
         display(table_to_process)
```

Figure 31: A part where all stop words are removed from the text.



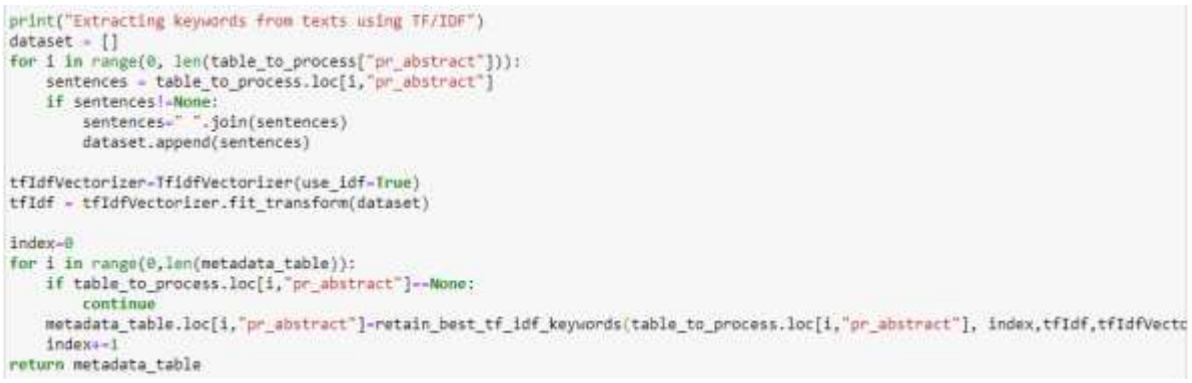
```
dist = nltk.FreqDist(words_corpus) #Creating a distribution of words' frequencies
grams=dist.most_common(1000) #Obtaining the most frequent words
bigrams = nltk.collocations.BigramAssocMeasures()
trigrams = nltk.collocations.TrigramAssocMeasures()

bigramFinder = nltk.collocations.BigramCollocationFinder.from_words(words_corpus)
trigramFinder = nltk.collocations.TrigramCollocationFinder.from_words(words_corpus)

print("Showing first",2000,"top-freqent words in the corpus")
grams = pd.DataFrame(grams) |
grams.index = range(1,len(grams)+1)
grams.columns = ["Word", "Frequency"]
display(grams)

bi_filter=7
print("Showing bigrams in the corpus found by Pointwise Mutual Information method")
print("Applying frequency filter: a bigramm occurs more than",bi_filter,"times")
bigramFinder.apply_freq_filter(bi_filter)
bigramPMITable = pd.DataFrame(list(bigramFinder.score_ngrams(bigrams.pmi)), columns=['bigram','PMI']).sort_values(by='PMI', as
bigramPMITable["bigram"]=bigramPMITable["bigram"].apply(lambda x: ' '.join(x))
display(bigramPMITable)

tri_filter=5
print("Showing trigrams in the corpus found by Pointwise Mutual Information method")
print("Applying frequency filter: a trigramm occurs more than",tri_filter,"times")
trigramFinder.apply_freq_filter(tri_filter)
trigramPMITable = pd.DataFrame(list(trigramFinder.score_ngrams(trigrams.pmi)), columns=['trigram','PMI']).sort_values(by='PMI'
trigramPMITable["trigram"]=trigramPMITable["trigram"].apply(lambda x: ' '.join(x))
display(trigramPMITable)
```

Figure 32: A part where grams, bigrams and trigrams are selected.



```
keyword_processor = KeywordProcessor()
textrank_keyword_processor = KeywordProcessor()

gram_dict.update(bigramPMIDict)
bigramPMIDict.update(trigramPMIDict)

  print(gram_dict)
print("Extracting keywords from texts using Pointwise Mutual Information method and TextRank")
text_rank_key_words=dict()
for i in range(0, len(table_to_process)):
    sentences=table_to_process.loc[i,"pr_abstract"]
    if sentences!=None:
        keywords=get_keywords_by_textrank(sentences)
        if keywords!=None:
            text_rank_key_words.update(keywords)
            print("Text",i,"- Done")

for keyword in gram_dict.keys():
    parts=keyword.split()
    parts="_".join(parts)
    keyword_processor.add_keyword(keyword,parts)

for keyword in text_rank_key_words.keys():
    parts=keyword.split()
    parts="_".join(parts)
    textrank_keyword_processor.add_keyword(keyword,parts)
```

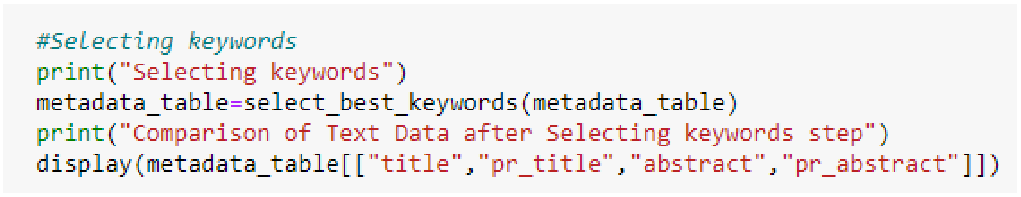
Figure 33: A part where all keywords are extracted from texts by both extracting methods.



```
print("Extracting keywords from texts using TF/IDF")
dataset = []
for i in range(0, len(table_to_process["pr_abstract"])):
    sentences = table_to_process.loc[i,"pr_abstract"]
    if sentences!=None:
        sentences=" ".join(sentences)
        dataset.append(sentences)

tfIdfVectorizer=TfidfVectorizer(use_idf=True)
tfIdf = tfIdfVectorizer.fit_transform(dataset)

index=0
for i in range(0,len(metadata_table)):
    if table_to_process.loc[i,"pr_abstract"]==None:
        continue
    metadata_table.loc[i,"pr_abstract"]=retain_best_tf_idf_keywords(table_to_process.loc[i,"pr_abstract"], index,tfIdf,tfIdfVect
    index+=1
return metadata_table
```

Figure 34: A part where the TF/IDF method is applied to obtained texts.

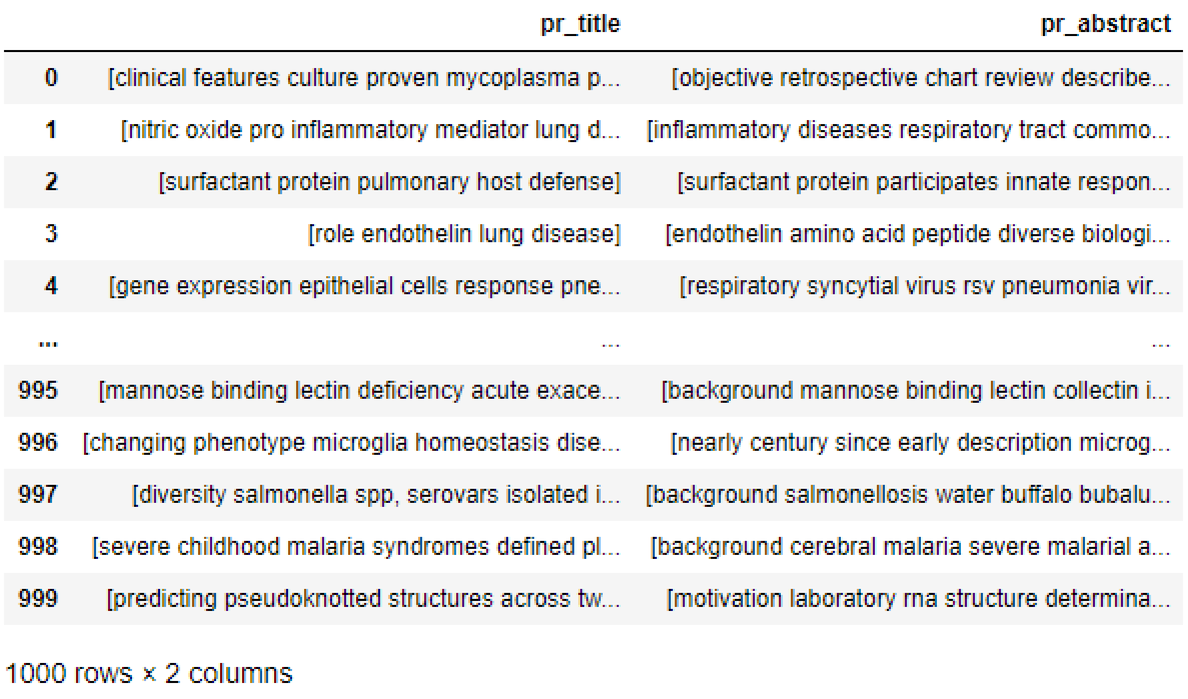
As it is depicted on the figure 35, the “keywords selection” step is performed.



```
#Selecting keywords
print("Selecting keywords")
metadata_table=select_best_keywords(metadata_table)
print("Comparison of Text Data after Selecting keywords step")
display(metadata_table[["title","pr_title","abstract","pr_abstract"]])
```

Figure 35: A part of the code in the function “run_programme” that does the “keywords selection” step.

A table with text data after removing of stop words is shown on the figure 36 below. We can notice that text data do not contain any stop words such as “don’t”, “of” or “this”.

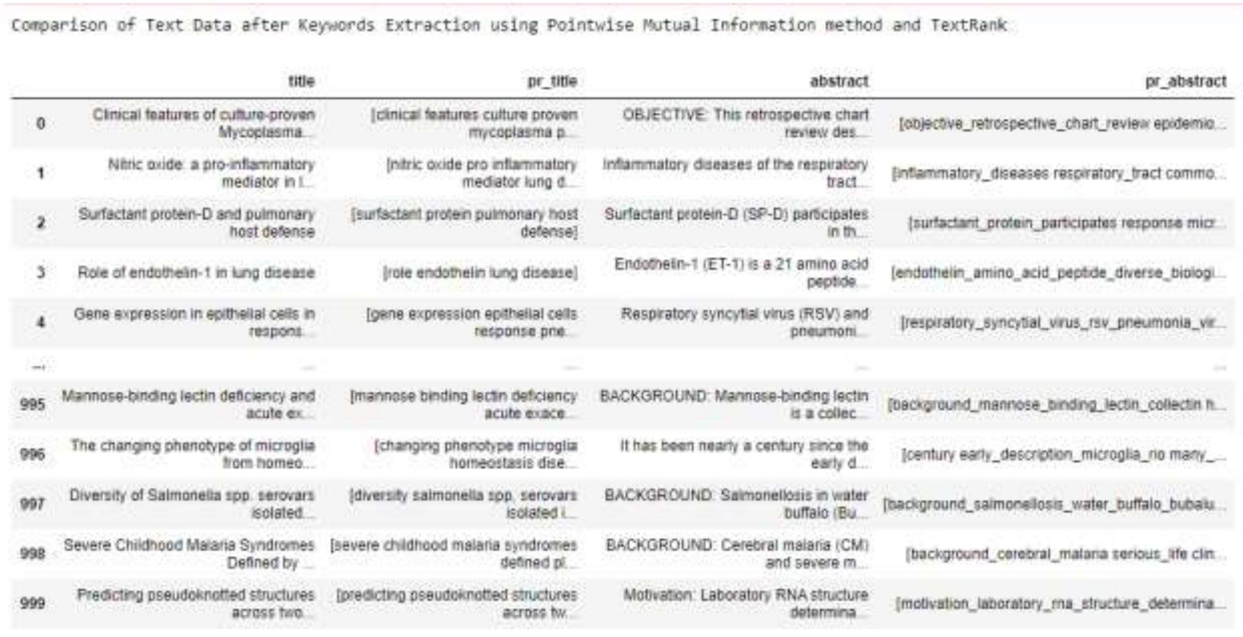


	pr_title	pr_abstract
0	[clinical features culture proven mycoplasma p...	[objective retrospective chart review describe...
1	[nitric oxide pro inflammatory mediator lung d...	[inflammatory diseases respiratory tract commo...
2	[surfactant protein pulmonary host defense]	[surfactant protein participates innate respon...
3	[role endothelin lung disease]	[endothelin amino acid peptide diverse biologi...
4	[gene expression epithelial cells response pne...	[respiratory syncytial virus rsv pneumonia vir...
...	...	...
995	[mannose binding lectin deficiency acute exace...	[background mannose binding lectin collectin i...
996	[changing phenotype microglia homeostasis dise...	[nearly century since early description microg...
997	[diversity salmonella spp, serovars isolated i...	[background salmonellosis water buffalo bubalu...
998	[severe childhood malaria syndromes defined pl...	[background cerebral malaria severe malarial a...
999	[predicting pseudoknotted structures across tw...	[motivation laboratory rna structure determina...

1000 rows × 2 columns

Figure 36: A table that demonstrates a text data without any stop words

On the figure 37 below a comparative table that demonstrates results of keywords selection by PMI and TextRank is depicted. We can notice that some keywords in the column "pr_abstract" are surrounded by a sign "_" to keep their structure for TF/IDF.

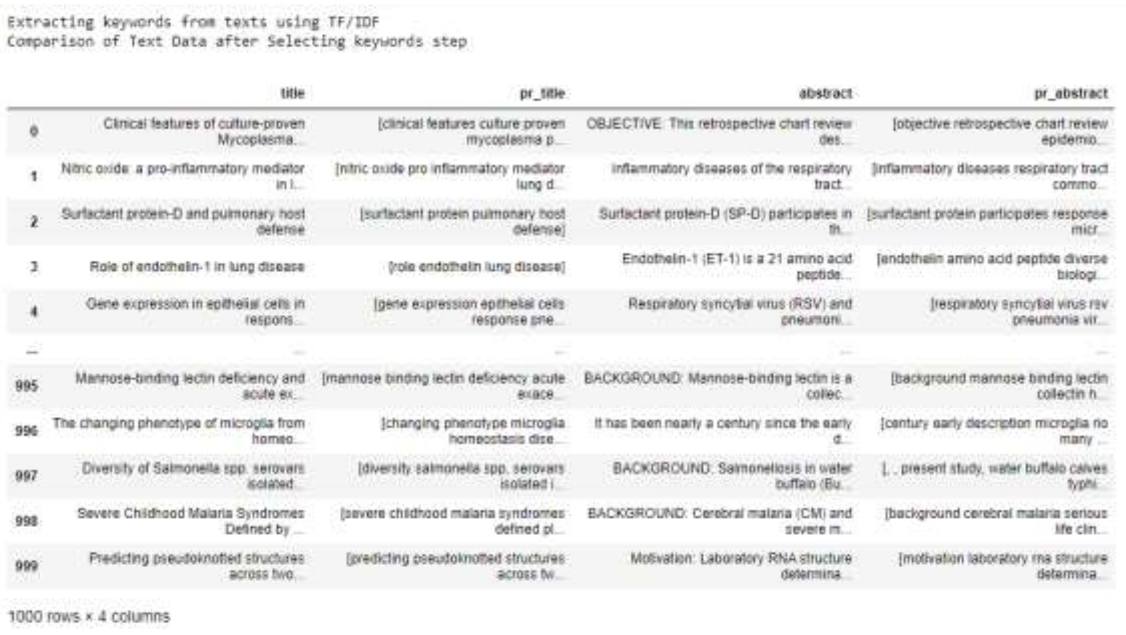


Comparison of Text Data after Keywords Extraction using Pointwise Mutual Information method and TextRank

	title	pr_title	abstract	pr_abstract
0	Clinical features of culture-proven Mycoplasma...	[clinical features culture proven mycoplasma p...	OBJECTIVE: This retrospective chart review des...	[objective_retrospective_chart_review epidemio...
1	Nitric oxide: a pro-inflammatory mediator in l...	[nitric oxide pro inflammatory mediator lung d...	Inflammatory diseases of the respiratory tract...	[inflammatory_diseases respiratory_tract commo...
2	Surfactant protein-D and pulmonary host defense	[surfactant protein pulmonary host defense]	Surfactant protein-D (SP-D) participates in th...	[surfactant_protein_participates response micr...
3	Role of endothelin-1 in lung disease	[role endothelin lung disease]	Endothelin-1 (ET-1) is a 21 amino acid peptide...	[endothelin_amino_acid_peptide_diverse_biologi...
4	Gene expression in epithelial cells in respons...	[gene expression epithelial cells response pne...	Respiratory syncytial virus (RSV) and pneumoni...	[respiratory_syncytial_virus_rsv_pneumonia_vir...
...	...	...	...	...
995	Mannose-binding lectin deficiency and acute ex...	[mannose binding lectin deficiency acute exace...	BACKGROUND: Mannose-binding lectin is a collec...	[background_mannose_binding_lectin_collectin h...
996	The changing phenotype of microglia from homeo...	[changing phenotype microglia homeostasis dise...	It has been nearly a century since the early d...	[century early_description_microglia_rio many_...
997	Diversity of Salmonella spp. serovars isolated...	[diversity salmonella spp, serovars isolated i...	BACKGROUND: Salmonellosis in water buffalo (Bu...	[background_salmonellosis_water_buffalo_bubalu...
998	Severe Childhood Malaria Syndromes Defined by ...	[severe childhood malaria syndromes defined pl...	BACKGROUND: Cerebral malaria (CM) and severe m...	[background_cerebral_malaria serious_life clin...
999	Predicting pseudoknotted structures across two...	[predicting pseudoknotted structures across tw...	Motivation: Laboratory RNA structure determina...	[motivation_laboratory_rna_structure_determina...

Figure 37: A comparative table of differences between original texts and texts with only keywords.

A result of applying the TF/IDF keywords extraction method is exhibited on the figure 38. We can notice that all "_" signs have already disappeared, because there is no need to keep them for further steps.



```
Extracting keywords from texts using TF/IDF
Comparison of Text Data after Selecting keywords step
```

	title	pr_title	abstract	pr_abstract
0	Clinical features of culture-proven Mycoplasma...	[clinical features culture proven mycoplasma p...	OBJECTIVE: This retrospective chart review des...	[objective retrospective chart review epidemio...
1	Nitric oxide: a pro-inflammatory mediator in l...	[nitric oxide pro inflammatory mediator lung d...	Inflammatory diseases of the respiratory tract...	[inflammatory diseases respiratory tract commo...
2	Surfactant protein-D and pulmonary host defense	[surfactant protein pulmonary host defense]	Surfactant protein-D (SP-D) participates in th...	[surfactant protein participates response micr...
3	Role of endothelin-1 in lung disease	[role endothelin lung disease]	Endothelin-1 (ET-1) is a 21 amino acid peptide...	[endothelin amino acid peptide diverse biologi...
4	Gene expression in epithelial cells in respons...	[gene expression epithelial cells response pne...	Respiratory syncytial virus (RSV) and pneumoni...	[respiratory syncytial virus rsv pneumonia vir...
...	...	...	...	...
995	Mannose-binding lectin deficiency and acute ex...	[mannose binding lectin deficiency acute exace...	BACKGROUND: Mannose-binding lectin is a collec...	[background mannose binding lectin collectin h...
996	The changing phenotype of microglia from homeo...	[changing phenotype microglia homeostasis dise...	It has been nearly a century since the early d...	[century early description microglia rio many ...
997	Diversity of Salmonella spp. serovars isolated...	[diversity salmonella spp. serovars isolated i...	BACKGROUND: Salmonellosis in water buffalo (Bu...	[, . present study, water buffalo calves typhi...
998	Severe Childhood Malaria Syndromes Defined by ...	[severe childhood malaria syndromes defined pl...	BACKGROUND: Cerebral malaria (CM) and severe m...	[background cerebral malaria serious life clin...
999	Predicting pseudoknotted structures across two...	[predicting pseudoknotted structures across tw...	Motivation: Laboratory RNA structure determina...	[motivation laboratory rna structure determina...

1000 rows × 4 columns

Figure 38: A comparative table of differences between normalised text data and not normalised text data

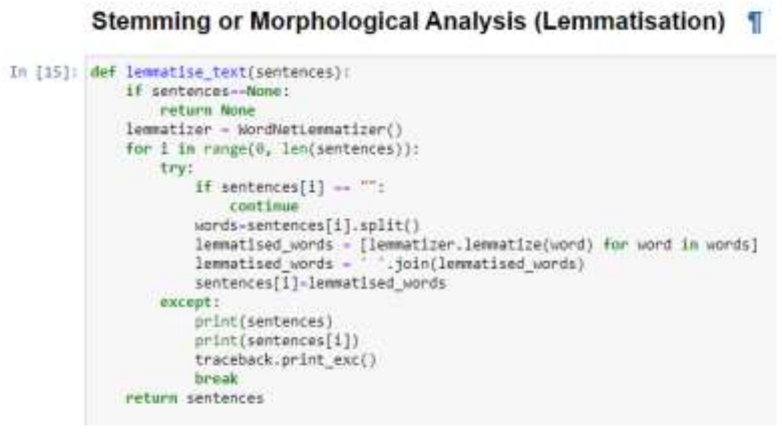
Stemming or Morphological Analysis

Include here the details of how you did this step including any issue that you had and how did you face it. Present examples for each of the aspects where this step went well. Also include examples for when it when wrong and how you could solve it. You may include screenshots to clarify.

Initially, implementing of a certain stemming technique was a rather fine decision because it is rather simple and removes word endings for most of words. However, the main problem that occurred in this project is that this method merely cuts a word ending thereby corrupting a word itself and making it unsuitable for a search.

To avoid this problem, there was made a decision to implement a lemmatisation technique. A process of lemmatisation allowed to extract lemmas from words thereby preserving their meaning. Thus, it was a much safer method to process words than a stemming and more efficient.

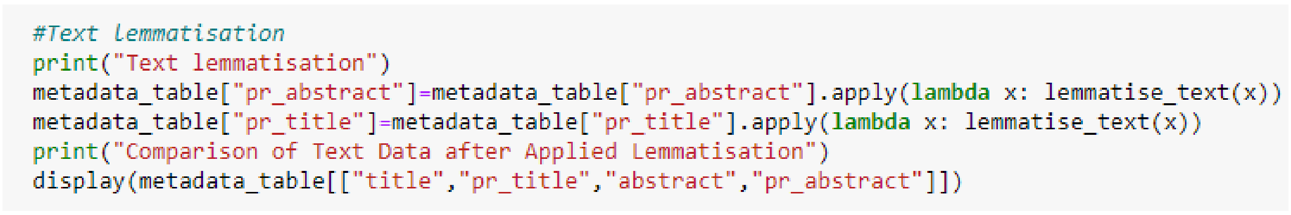
A function that does lemmatisation of texts can be seen on the figure 39 below. Also, it is called in the code part of the function “run_program” that is depicted on the figure 40.



Stemming or Morphological Analysis (Lemmatisation) ¶

```
In [15]: def lemmatise_text(sentences):
             if sentences==None:
                 return None
             lemmatizer = WordNetLemmatizer()
             for i in range(0, len(sentences)):
                 try:
                     if sentences[i] == "":
                         continue
                     words=sentences[i].split()
                     lemmatised_words = [lemmatizer.lemmatize(word) for word in words]
                     lemmatised_words = ' '.join(lemmatised_words)
                     sentences[i]=lemmatised_words
                 except:
                     print(sentences)
                     print(sentences[i])
                     traceback.print_exc()
                     break
             return sentences
```

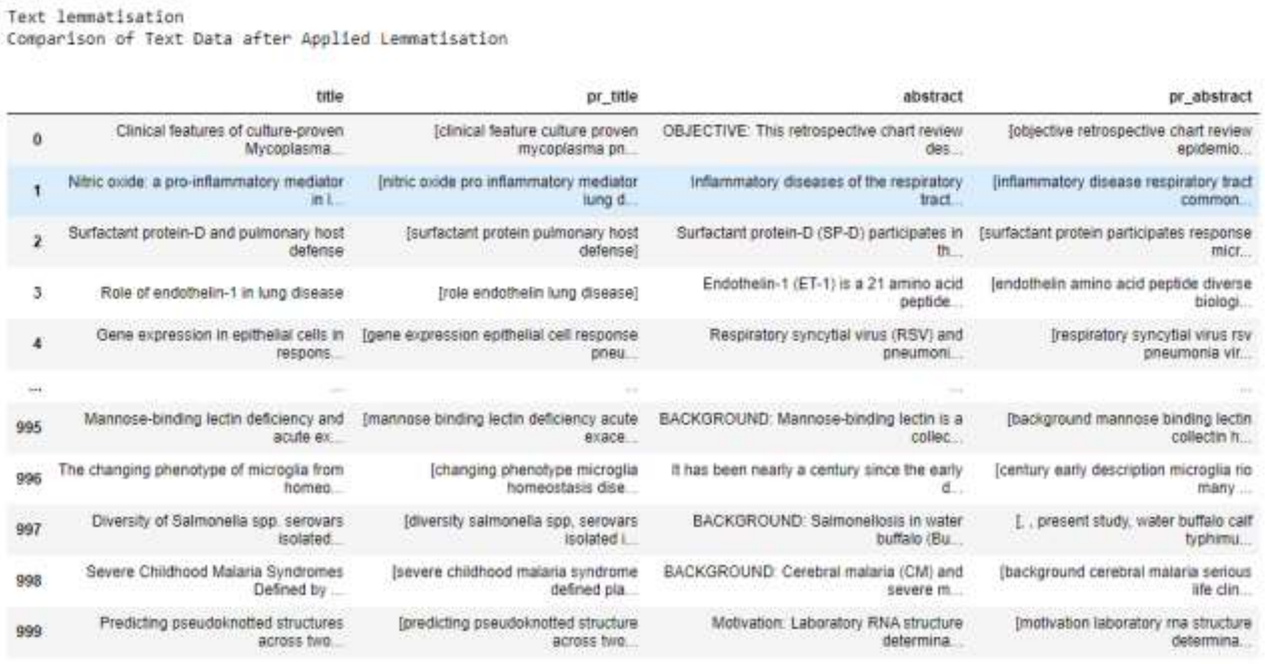
Figure 39: A function that does lemmatisation of texts



```
#Text Lemmatisation
print("Text lemmatisation")
metadata_table["pr_abstract"]=metadata_table["pr_abstract"].apply(lambda x: lemmatise_text(x))
metadata_table["pr_title"]=metadata_table["pr_title"].apply(lambda x: lemmatise_text(x))
print("Comparison of Text Data after Applied Lemmatisation")
display(metadata_table[["title","pr_title","abstract","pr_abstract"]])
```

Figure 40: A part of the code of the function "run_program" that does the "Stemming" step

According to the table 41 below, some changes after applied lemmatisation to texts is clearly demonstrated. For example, a word "features" was transformed to the word "feature".



```
Text lemmatisation
Comparison of Text Data after Applied Lemmatisation
```

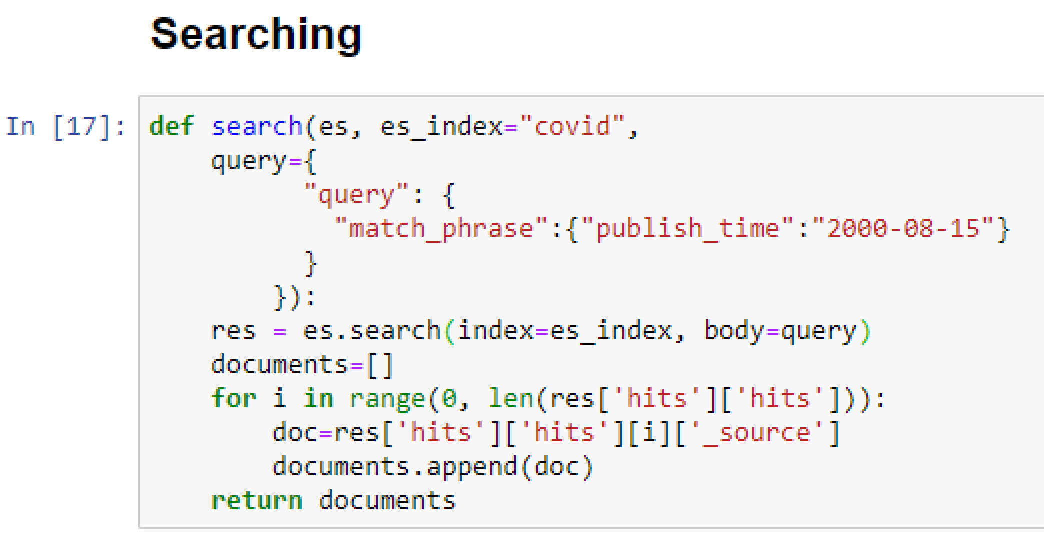
	title	pr_title	abstract	pr_abstract
0	Clinical features of culture-proven Mycoplasma...	[clinical feature culture proven mycoplasma pn...	OBJECTIVE: This retrospective chart review des...	[objective retrospective chart review epidemio...
1	Nitric oxide: a pro-inflammatory mediator in l...	[nitric oxide pro inflammatory mediator lung d...	Inflammatory diseases of the respiratory tract...	[inflammatory disease respiratory tract common...
2	Surfactant protein-D and pulmonary host defense	[surfactant protein pulmonary host defense]	Surfactant protein-D (SP-D) participates in th...	[surfactant protein participates response micr...
3	Role of endothelin-1 in lung disease	[role endothelin lung disease]	Endothelin-1 (ET-1) is a 21 amino acid peptide...	[endothelin amino acid peptide diverse biologi...
4	Gene expression in epithelial cells in respons...	[gene expression epithelial cell response pneu...	Respiratory syncytial virus (RSV) and pneumoni...	[respiratory syncytial virus rsv pneumonia vir...
...	...	...	...	...
995	Mannose-binding lectin deficiency and acute ex...	[mannose binding lectin deficiency acute exace...	BACKGROUND: Mannose-binding lectin is a collec...	[background mannose binding lectin collectin h...
996	The changing phenotype of microglia from homeo...	[changing phenotype microglia homeostasis dise...	It has been nearly a century since the early d...	[century early description microglia rio many ...
997	Diversity of Salmonella spp. serovars isolated...	[diversity salmonella spp, serovars isolated i...	BACKGROUND: Salmonellosis in water buffalo (Bu...	[, , present study, water buffalo calf typhimu...
998	Severe Childhood Malaria Syndromes Defined by ...	[severe childhood malaria syndrome defined pla...	BACKGROUND: Cerebral malaria (CM) and severe m...	[background cerebral malaria serious life clin...
999	Predicting pseudoknotted structures across two...	[predicting pseudoknotted structure across two...	Motivation: Laboratory RNA structure determina...	[motivation laboratory rna structure determina...

Figure 41: A comparative table of differences between lemmatised text data and not lemmatised text.

Searching

Include here the details of how you did this step including any issue that you had and how did you face it. You may include screenshots to clarify.

Having an established connection with an ElasticSearch engine, a name of index of a document collection and a query to process, it is quite straightforward and convenient to interact with ElasticSearch using "es.search()" function that is represented on the figure 42 below.



Searching

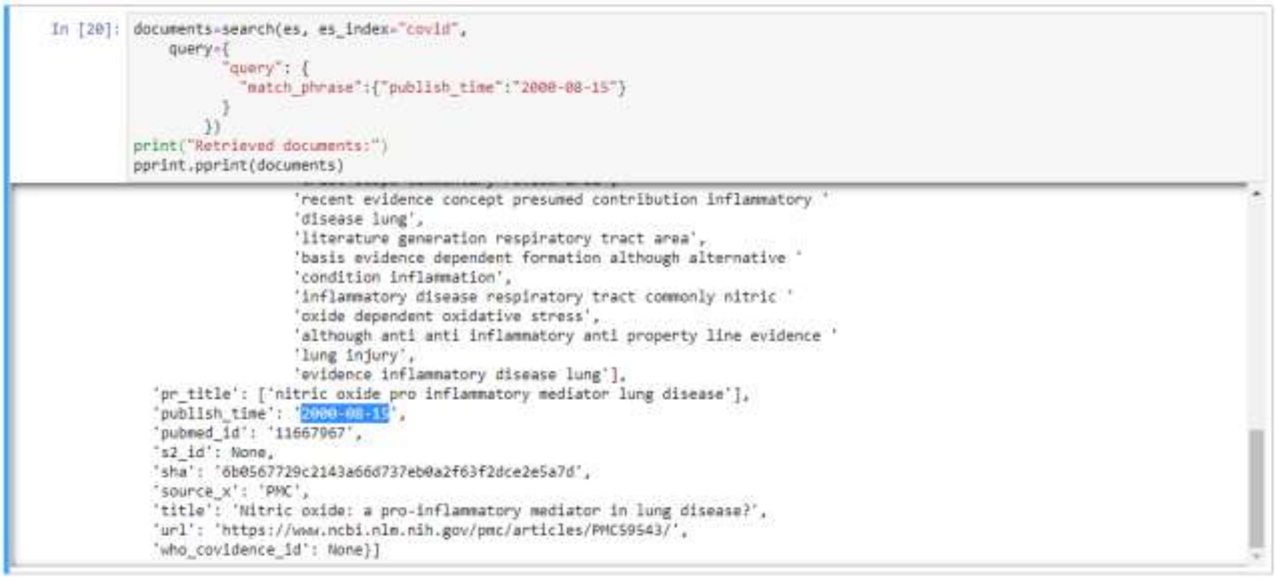
```
In [17]: def search(es, es_index="covid",
             query={
                    "query": {
                       "match_phrase":{"publish_time":"2000-08-15"}
                    }
                 }):
             res = es.search(index=es_index, body=query)
             documents=[]
             for i in range(0, len(res['hits']['hits'])):
                 doc=res['hits']['hits'][i]['_source']
                 documents.append(doc)
             return documents
```

Figure 42: A description of an implemented function “search” that can send queries to an ElasticSearch engine.

Although this, function appears to be trivial, however the main issue was to how to access retrieved documents and represent their content in a smooth and convenient way.

Having considered a JSON format of retrieved documents, there was found a way to extract those documents using properties of a dictionary type in Python. As it is depicted on the figure above, it took several levels of depth to return found documents.

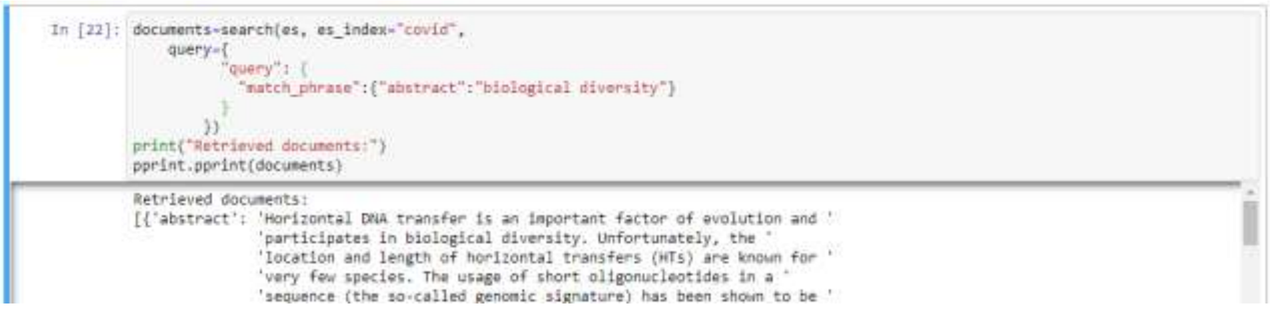
Having done that, a user can perform a search by specifying a desired query. On figures 43,44,45,46 below it can be clearly seen that using Python a user can make up any desired query and obtain a relevant result.



```
In [20]: documents=search(es, es_index="covid",
             query={
                 "query": {
                   "match_phrase":{"publish_time":"2000-08-15"}
                 }
              })
         print("Retrieved documents:")
         pprint.pprint(documents)
```

```
                         'recent evidence concept presumed contribution inflammatory '
                         'disease lung',
                         'literature generation respiratory tract area',
                         'basis evidence dependent formation although alternative '
                         'condition inflammation',
                         'inflammatory disease respiratory tract commonly nitric '
                         'oxide dependent oxidative stress',
                         'although anti anti inflammatory anti property line evidence '
                         'lung injury',
                         'evidence inflammatory disease lung'],
  'pr_title': ['nitric oxide pro inflammatory mediator lung disease'],
  'publish_time': '2000-08-15',
  'pubmed_id': '11667967',
  's2_id': None,
  'sha': '6b0567729c2143a66d737eb0a2f63f2dce2e5a7d',
  'source_x': 'PMC',
  'title': 'Nitric oxide: a pro-inflammatory mediator in lung disease?',
  'url': 'https://www.ncbi.nlm.nih.gov/pmc/articles/PMC59543/',
  'who_covidence_id': None}]
```

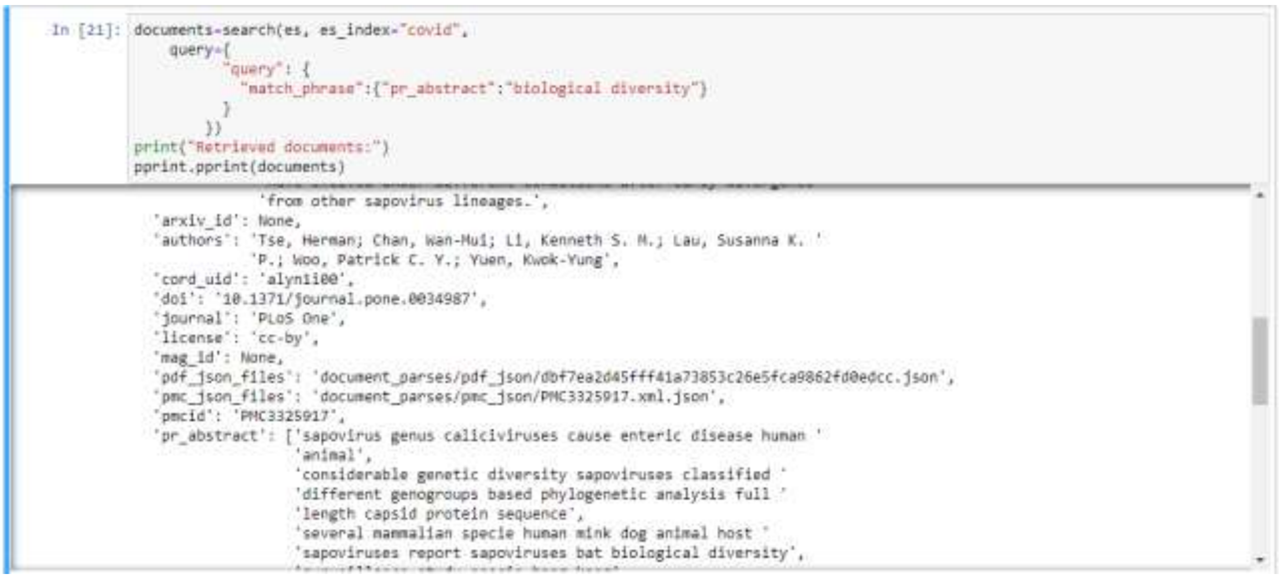
Figure 43 A description of an implemented function “search” that can send queries to an ElasticSearch engine.



```
In [22]: documents=search(es, es_index="covid",
             query={
                    "query": {
                      "match_phrase":{"abstract":"biological diversity"}
                    }
                 })
         print("Retrieved documents:")
         pprint.pprint(documents)
```

```
Retrieved documents:
[{'abstract': 'Horizontal DNA transfer is an important factor of evolution and '
              'participates in biological diversity. Unfortunately, the '
              'location and length of horizontal transfers (HTs) are known for '
              'very few species. The usage of short oligonucleotides in a '
              'sequence (the so-called genomic signature) has been shown to be '
```

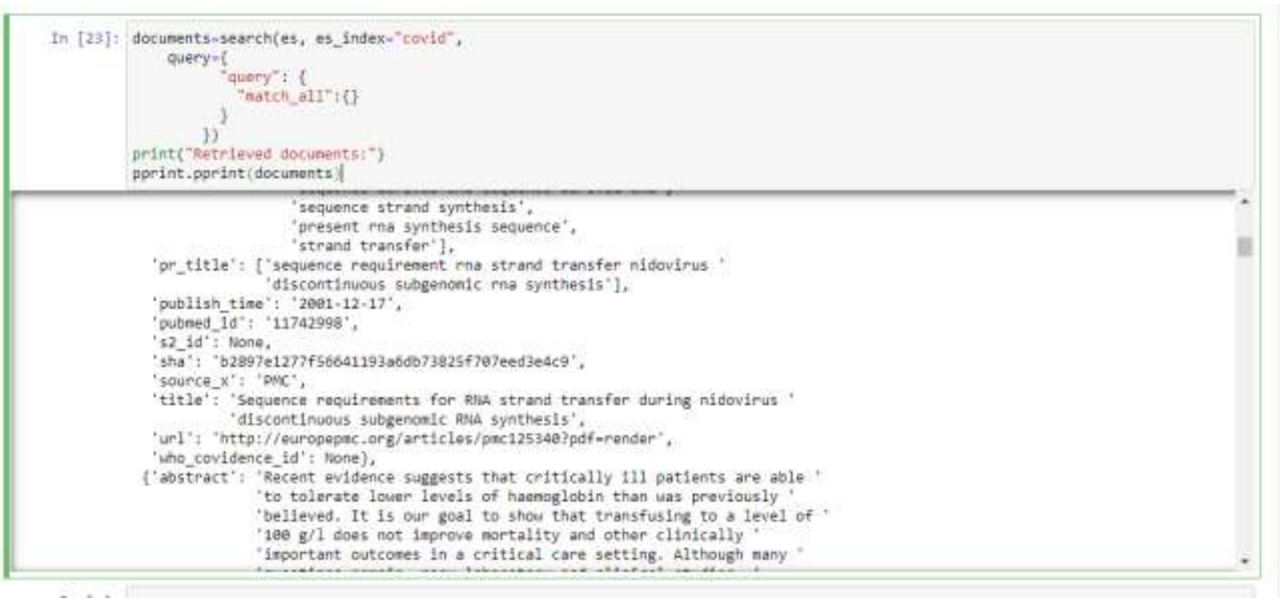
Figure 44: A description of an implemented function “search” that can send queries to an ElasticSearch engine.



```
In [21]: documents=search(es, es_index="covid",
             query={
                    "query": {
                      "match_phrase":{"pr_abstract":"biological diversity"}
                    }
                 })
         print("Retrieved documents:")
         pprint.pprint(documents)
```

```
                'from other sapovirus lineages.',
  'arxiv_id': None,
  'authors': 'Tse, Herman; Chan, Wan-Mui; Li, Kenneth S. M.; Lau, Susanna K. '
             'P.; Woo, Patrick C. Y.; Yuen, Kwok-Yung',
  'cord_uid': 'alyn1i00',
  'doi': '10.1371/journal.pone.0034987',
  'journal': 'PLoS One',
  'license': 'cc-by',
  'mag_id': None,
  'pdf_json_files': 'document_parses/pdf_json/dbf7ea2d45fff41a73853c26e5fca9862fd0edcc.json',
  'pmc_json_files': 'document_parses/pmc_json/PMC3325917.xml.json',
  'pmcid': 'PMC3325917',
  'pr_abstract': ['sapovirus genus caliciviruses cause enteric disease human '
                  'animal',
                  'considerable genetic diversity sapoviruses classified '
                  'different genogroups based phylogenetic analysis full '
                  'length capsid protein sequence',
                  'several mammalian specie human mink dog animal host '
                  'sapoviruses report sapoviruses bat biological diversity',
```

Figure 45: A description of an implemented function “search” that can send queries to an ElasticSearch engine.

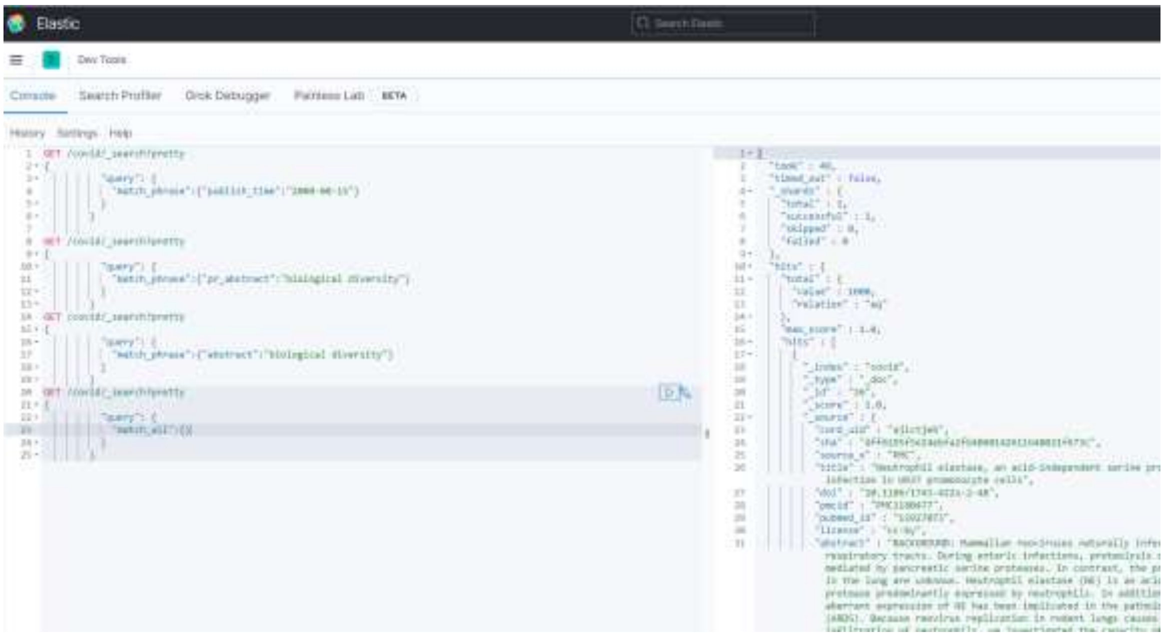


```
In [23]: documents=search(es, es_index="covid",
             query={
                 "query": {
                   "match_all":{}
                 }
             })
         print("Retrieved documents:")
         pprint.pprint(documents)
```

```
                         'sequence strand synthesis',
                         'present rna synthesis sequence',
                         'strand transfer'],
          'pr_title': ['sequence requirement rna strand transfer nidovirus '
                       'discontinuous subgenomic rna synthesis'],
          'publish_time': '2001-12-17',
          'pubmed_id': '11742998',
          's2_id': None,
          'sha': 'b2897e1277f56641193a6db73825f707eed3e4c9',
          'source_x': 'PMC',
          'title': 'Sequence requirements for RNA strand transfer during nidovirus '
                   'discontinuous subgenomic RNA synthesis',
          'url': 'http://europepmc.org/articles/pmc125340?pdf=render',
          'who_covidence_id': None},
         {'abstract': 'Recent evidence suggests that critically ill patients are able '
                      'to tolerate lower levels of haemoglobin than was previously '
                      'believed. It is our goal to show that transfusing to a level of '
                      '100 g/l does not improve mortality and other clinically '
                      'important outcomes in a critical care setting. Although many '
```

Figure 46: A description of an implemented function “search” that can send queries to an ElasticSearch engine.

Also, a user can open a Kibana to obtain a more convenient and powerful environment to perform a search. An example of work with Kibana can be seen on the figure 47 below.



```
GET /covid/_search?pretty
{
      "query": {
        "match_phrase":{"publish_time":"2000-06-15"}
      }
    }

GET /covid/_search?pretty
{
      "query": {
        "match_phrase":{"pr_abstract":"biological diversity"}
      }
    }
GET covid/_search?pretty
{
      "query": {
        "match_phrase":{"abstract":"biological diversity"}
      }
    }
GET /covid/_search?pretty
{
      "query": {
        "match_all":{}
      }
    }
```

```
{
  "took" : 40,
  "timed_out" : false,
  "_shards" : {
    "total" : 1,
    "successful" : 1,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : {
      "value" : 1000,
      "relation" : "eq"
    },
    "max_score" : 1.0,
    "hits" : [
      {
        "_index" : "covid",
        "_type" : "_doc",
        "_id" : "20",
        "_score" : 1.0,
        "_source" : {
          "cord_uid" : "ejlctje6",
          "sha" : "[illegible]",
          "source_x" : "PMC",
          "title" : "Neutrophil elastase, an acid-independent serine pr
            infection in [illegible] pneumocyte cells",
          "doi" : "10.1186/1743-422X-2-48",
          "pmcid" : "PMC1180477",
          "pubmed_id" : "15927073",
          "license" : "cc-by",
          "abstract" : "BACKGROUND: Mammalian reoviruses naturally infe
            respiratory tracts. During enteric infections, proteolysis
            mediated by pancreatic serine proteases. In contrast, the p
            in the lung are unknown. Neutrophil elastase (NE) is an aci
            protease predominantly expressed by neutrophils. In additio
            aberrant expression of NE has been implicated in the pathol
            (ARDS). Because reovirus replication in rodent lungs causes
            infiltration of neutrophils, we investigated the capacity o
```

Figure 47: Making up search queries in the Kibana GUI.