**MODULE NAME:** INFORMATION RETRIEVAL

**MODULE CODE:** 7071CEM

**MODULE PROFESSOR:** Seyed Mousavi

**STUDENT NAME:** Guddu Kumar Shah

**STUDENT ID:** 11402949

[Task 1: Search Engine 3](#_Toc1581251980)

[1. Crawler 3](#_Toc1749167041)

[1.1. Number of publications 3](#_Toc1236105790)

[1.2. Information collection 4](#_Toc1052641646)

[1.3. Pre-Processing 5](#_Toc249544730)

[1.4. Weekly Crawler 6](#_Toc1190863568)

[1.5. Brief explanation 6](#_Toc2067515099)

[2. Indexer 7](#_Toc1650816427)

[2.1. Creating an index 7](#_Toc877827405)

[2.2. Inverted Index 7](#_Toc1022975855)

[2.3. Updation of Index 7](#_Toc478707418)

[2.4. Sample output on index 7](#_Toc795601519)

[3. Query Processor 8](#_Toc417432371)

[3.1. Pre-processing of Query 8](#_Toc1587985065)

[3.2. Boolean Function 8](#_Toc802301826)

[3.3. Convert user query 8](#_Toc382361276)

[3.4. Ranked Retrieval 8](#_Toc237579802)

[3.5. Demonstration 8](#_Toc798400207)

[3.6. Brief Explanation 8](#_Toc864533599)

[4. GUI App 10](#_Toc1328122840)

[Task 2: Document Clustering 12](#_Toc1176126016)

[1. Collected Documents 12](#_Toc548869733)

[2. Clustering Method 12](#_Toc593883238)

[3. Type of Clustering used 12](#_Toc1497070240)

[4. Demonstration 13](#_Toc1659897761)

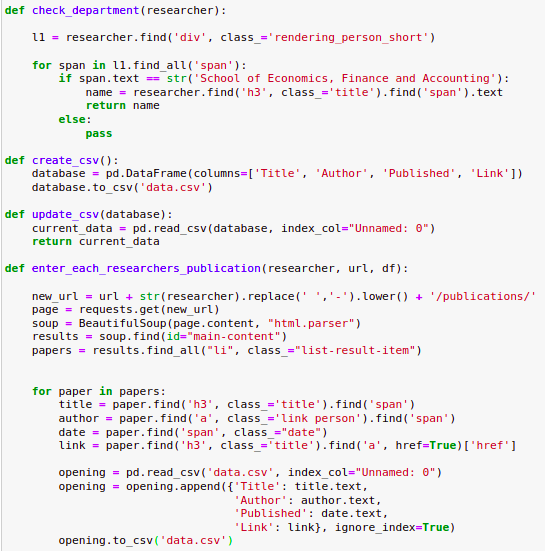
[5. Brief Explanation 14](#_Toc753975475)

# Task 1: Search Engine

## Crawler

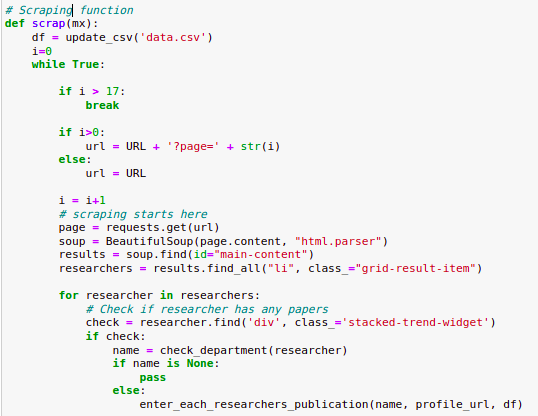
### Number of publications

In this task we are creating a search engine that can crawl data of coventry university publications from the department of school of economics finance and accounting. In the screenshot mentioned below we are setting up the seed url of the webpage from which we are going to crawl our data related to economics finance and accounting. In the next screenshot we are crawling the data in which our first function check\_department is used to check the publication belons to which department, and the next function is used to create a csv file to save all our publications data to a csv file for further pre-processing. In the next function we are crawling the data of each researcher data and their publication.



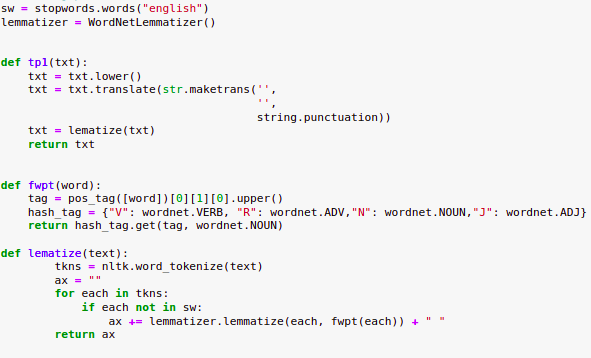
### Information collection

In this screenshot beloew we are scraping our data and saving it to a csv file named as data.csv. here we have used beautifulsoup library to parse the data from a html page and we have checked in this module if the researcher is having any publication if there are any publications of the researcher they are scraped and saved into our data.csv file.



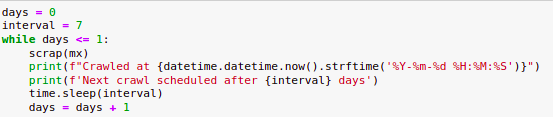
### Pre-Processing

In this task we have loaded our dataset and before passing the data to our indexer we will be performing some pre-processing steps on our data. The first pre-processing step we are going to use is to remove stopwords which are unnecessary words. The stopwords list was downloaded from nltk which is a library for text related task the next step is lemmatizing our data in which all the 3rd form of our data is converted to first form, for this purpose wordnetlemmatizer was used. In the next step we are lowercasing our entire dataframe and removing punctuations from our data, and the last step is to use pos tagging on our data.



### Weekly Crawler

After creating the crawler we have scheduled our crawler to automatically crawl our data as soon as we run our code or use it in real world problem. The crawler will scrap the data after every week. We have given the interval of 7 days to our crawler we can also provide the days if we want the crawler to run after every 3rd day. This crawler will scrap all the publication and update our csv file and we can further use it.



### Brief explanation

In this task we are creating a search engine for that we have crawled data of coventry university publications from the department of school of economics finance and accounting. After crawling the data multiple pre-processing steps are applied on the data and further it was indexed and query processor was applied to the data to use in our search engine.

## Indexer

### Creating an index

After performing the pre-processing on our data we have created an index of our data.

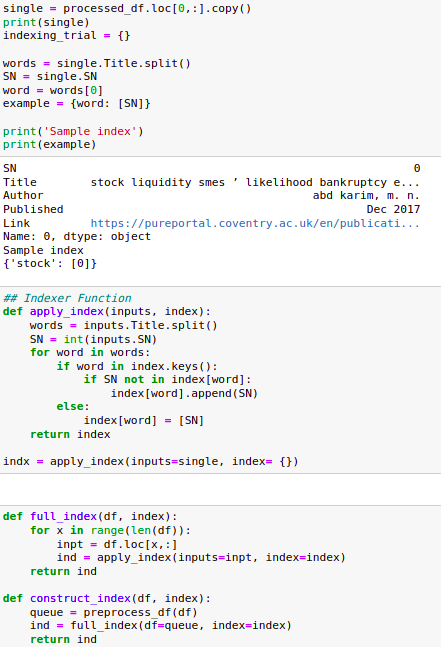
### Inverted Index

We have indexed our data into a json file named as index.json file. We have used inverted index as our index function in which we are creating a complete index of our data.

### Updation of Index

After creating the index as soon as the new data uploads the data in our index gets updated.

### Sample output on index



## Query Processor

### Pre-processing of Query

In this step we have lower cased our query and splitted it into multiple words for our search engine to understand the which publications to display as soon as the user gives an input.

### Boolean Function

We have used boolean function in our query processor part in this function we have used union and intersection on our query.

### Convert user query

We have used index instead of elastic search although our data is sorted using union and intersection in our boolean function.

### Ranked Retrieval

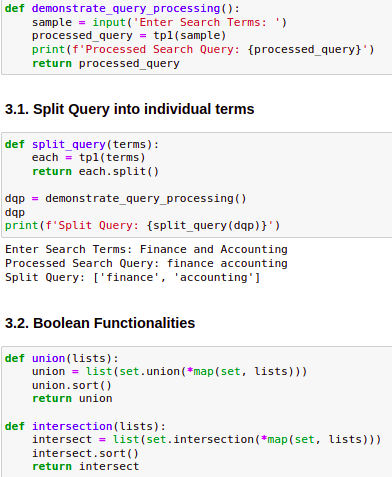
In this step our ranked retrieval returns the results on the basis of union and intersection of boolean function, Also the high and low rank are retrieved on the basis of boolean function.

### Demonstration

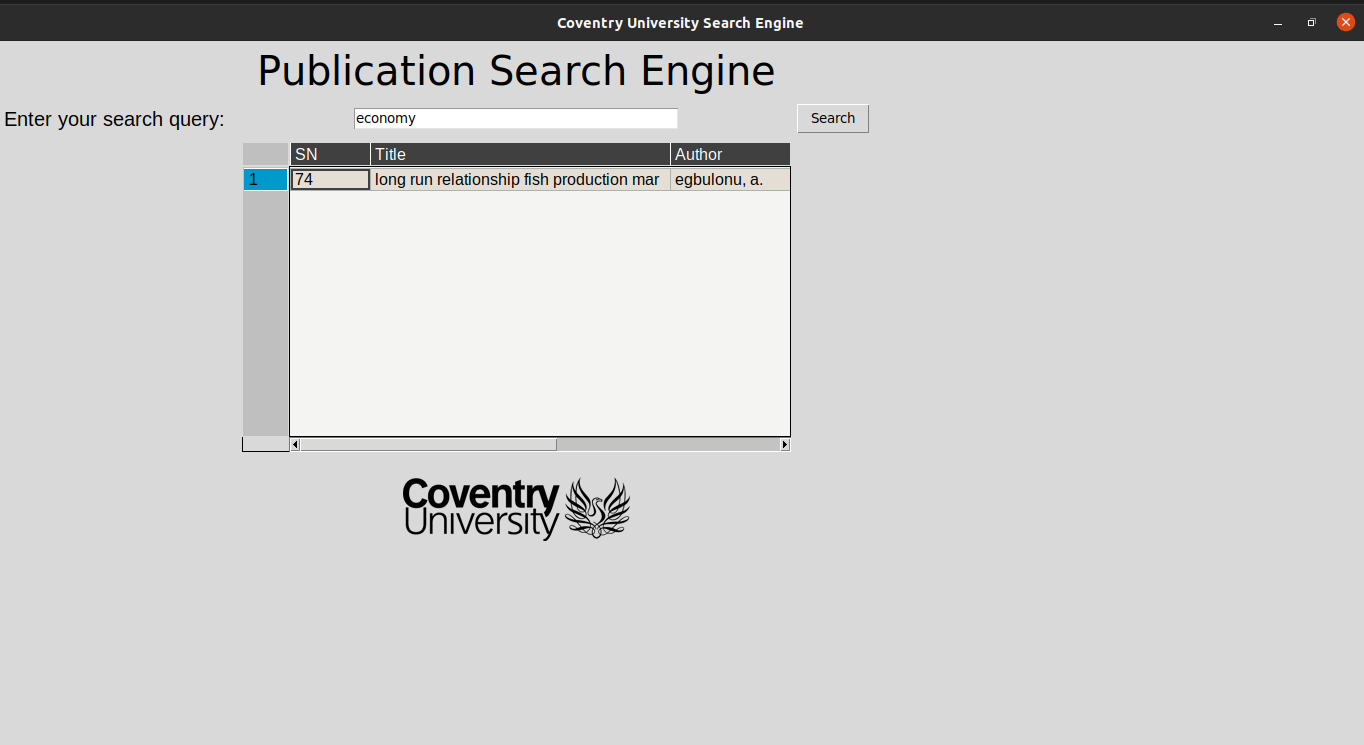
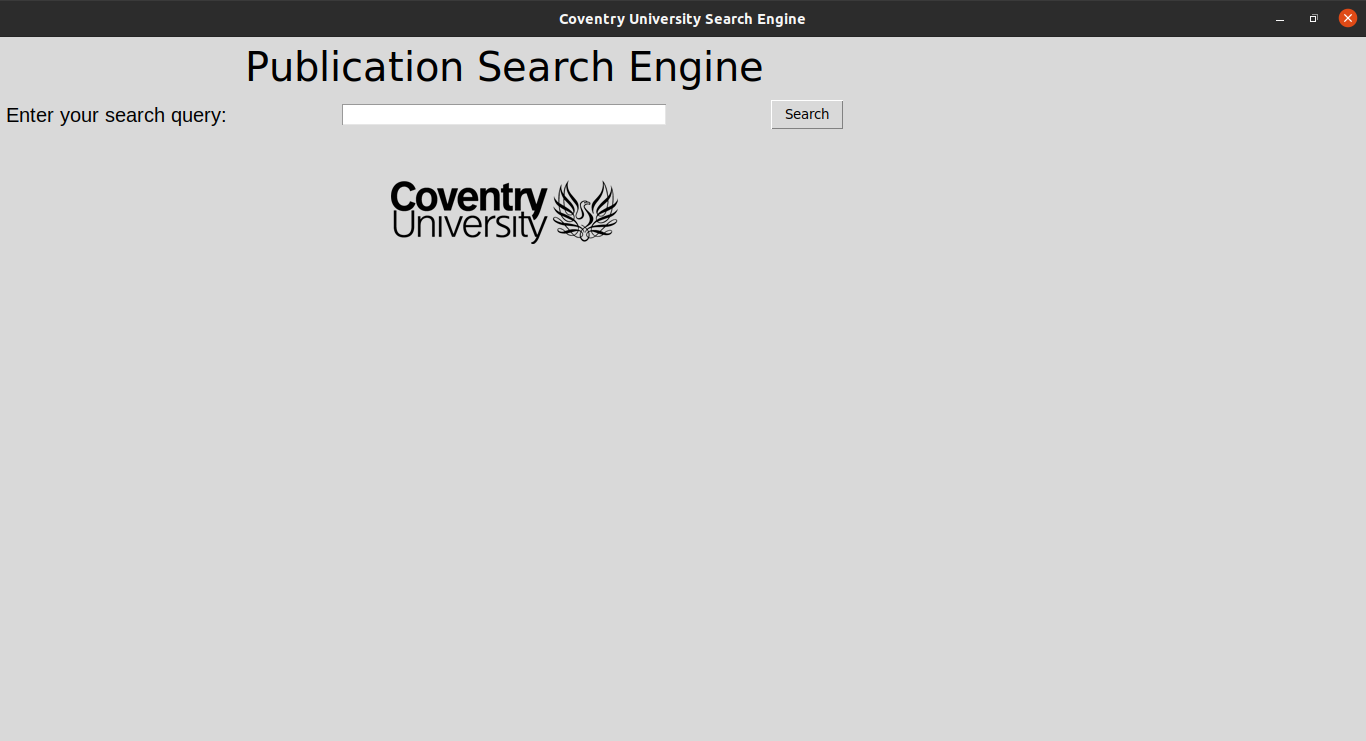
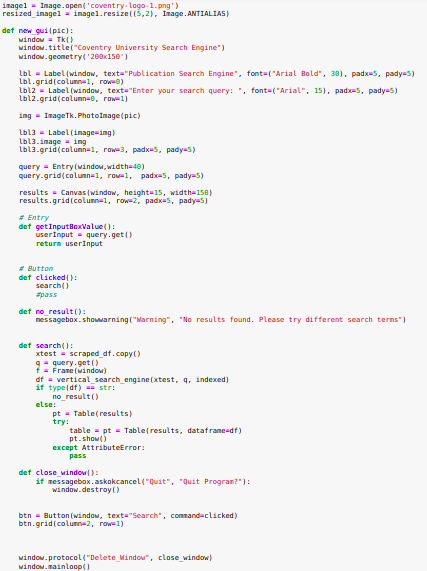
The demonstration of our query processor for our search engine is show in the screenshot below.

### Brief Explanation

For our query processor we have lower cased our query and splitted it into multiple words for our search engine to understand the which publications to display as soon as the user gives an input. After pre-processing boolean function was used in our query processor part in this function we have used union and intersection on our query. After boolean function ranked retrieval returns the results on the basis of union and intersection of boolean function, Also the high and low rank are retrieved on the basis of boolean function.



## GUI App



# Task 2: Document Clustering

### Collected Documents

There were 7768 total of documents collected for this task and were stored into a file named as Data\_Big.csv.

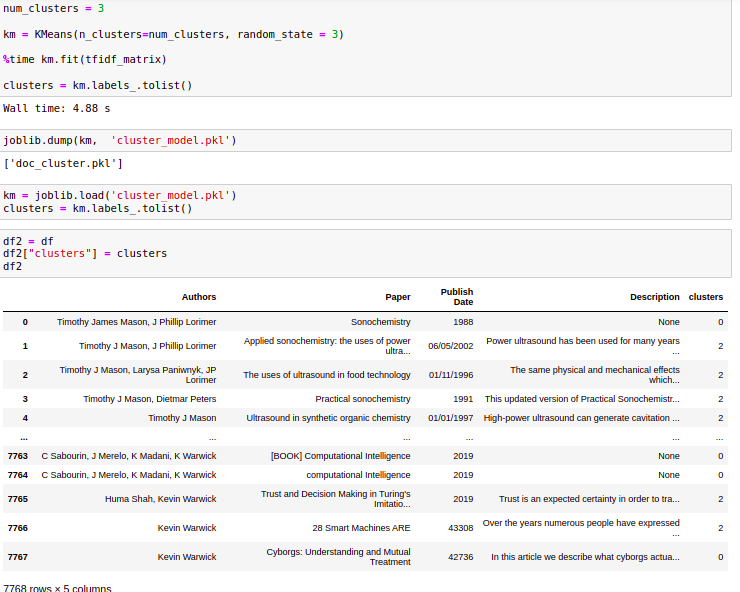
### Clustering Method

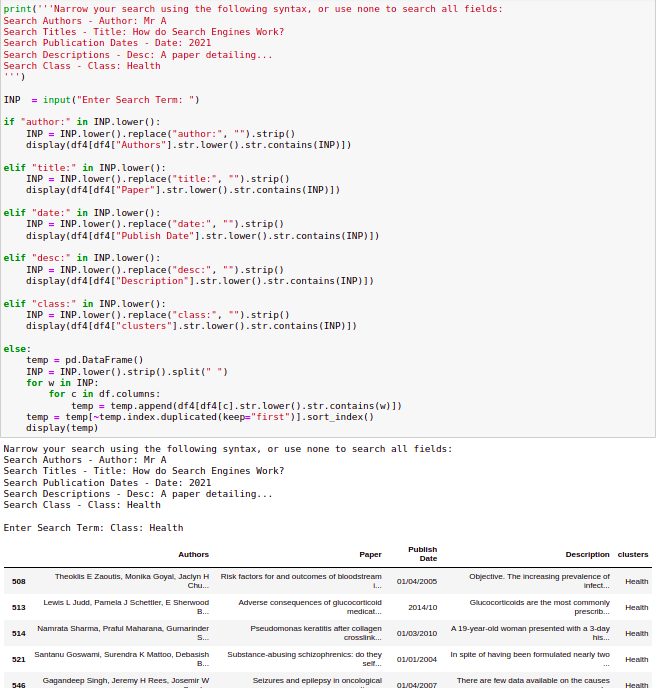
In this task we have used K-means clustering method and the clusters were set to 3 because we have to cluster between the documents related to sports, politics and health.

### Type of Clustering used

We have used hard clusting using K-means in which our clusters were set to 3 and we have used vectorization on our data before performing the clustering.

### Demonstration





### Brief Explanation

There were 7768 total of documents collected for this task and were stored into a file named as Data\_Big.csv. After loading the dataset there were various pre-processing steps performed on the data which includes lowercasing, tokenization and stemming. After performing the pre-processing steps vectorization was performed using Tf-Idf and then the clustering was performed using k-mean clustering model on the data. After clustering the documents a user can give input class in the runtime to display all the related documents of the class which a user has desired for and we can see the output in the screenshot above.

# Appendix

