**INFORMATION RETRIEVAL**

Table of Contents

[Part A (Software Implementation) 3](#_Toc130994671)

[Task 1: Search Engine 3](#_Toc130994672)

[1.1 Crawler 3](#_Toc130994673)

[1.2 Indexer 5](#_Toc130994674)

[1.3 Query Processor 7](#_Toc130994675)

[Task 2: Document Clustering 8](#_Toc130994676)

[Part B (Presentation) 14](#_Toc130994677)

[Introduction 14](#_Toc130994678)

[Task 1: Search Engine 14](#_Toc130994679)

[1.1 Crawler 14](#_Toc130994680)

[1.2 Indexer 16](#_Toc130994681)

[1.3 Query Processor 16](#_Toc130994682)

[Task 2: Document Clustering 17](#_Toc130994683)

[Conclusions 20](#_Toc130994684)

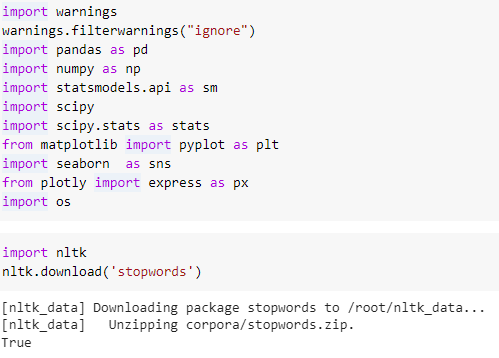
[References 21](#_Toc130994685)

# Part A (Software Implementation)

## Task 1: Search Engine

### 1.1 Crawler

Various information related to texts, repositories of documents, retrieval, storage, and organizations can be dealt with easily with the help of a software program called Information Retrieval or IR. The process of information retrieval has engaged various professional searchers, librarians, etc. In the case of access to information, the retrieval of information plays a dominant nature to satisfy the requirements of the user. In order to locate detailed information about any topic of interest, the users have been assisted by the IR but the answers have not been returned explicitly to the asked question.



**Figure 1.1.1: Importing of required libraries**

(Source: Acquired from Google collab)

Here in the above figure, the researcher has been mentioning the importing of different necessary libraries for this particular task. The library of pandas, numpy, matplotlib, etc. has been utilized for performing the programming for this given task.



**Figure 1.1.2: Importing of Word tokenizer**

(Source: Acquired from Google collab)

The “word tokenizer” is an essential tool for splitting a certain sentence or “text document” into several individual words or “tokens”. It is a method for text preprocessing that is used for applications dealing with “natural language processing”. There are numerous tokenizers that are built-in such as the “word\_tokenize” function used in the above figure.



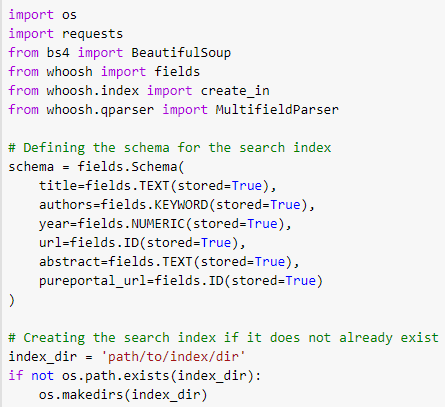
**Figure 1.1.3: Crawling of the search results**

(Source: Acquired from Googecollab)

The process of crawling is showcased in the above figure which is a process automated for the extraction of data from websites through programming code. It is a replication of systematic web browsing and collection of web page information.

### 1.2 Indexer

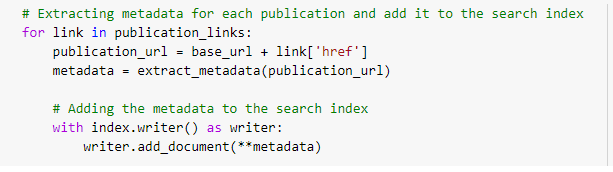
The action of acquiring the materials is known as Information Retrieval. Documentation of the information retrieval has been performed in an unstructured nature (Guo et al. 2020). The requirement of large collections of information from within the computer that has been stored there. It has been utilized in order to satisfy the text presented in the search bar. Millions of people have been involved in the process of retrieval of information every day whenever web search engine software has been used.



**Figure 1.2.1: Defining the schema**

(Source: Acquired from Google collab)

The above figure is the representation of the codes that are written to define the schema and to create the search index.



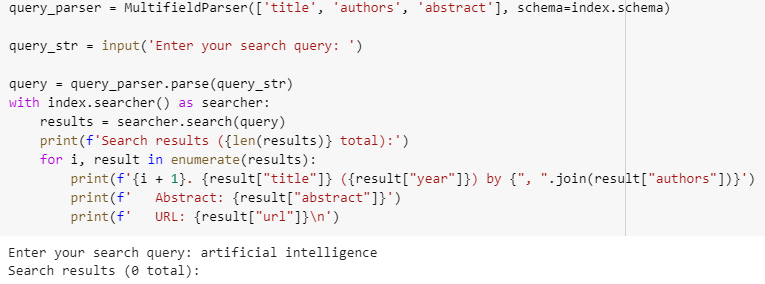
**Figure 1.2.2: Adding metadata to search index**

(Source: Acquired from Google collab)

Here the researcher has been performing the process of adding the metadata into the search index so that the efficacy of the whole program can be increased.

### 1.3 Query Processor

The location and existence of the documents that possess the required details can be notified by it. If a user is performing processing data from the retrieved documents, or performing filtration of collection of documents or browsing then they get various supports because of the information retrieval process (Lin, 2022). The data that has been stored in millions of computer systems has been searched by this system.



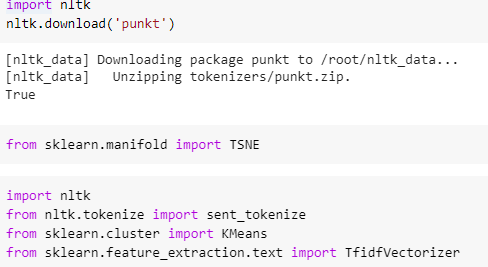
**Figure 1.3.1: Defining query parser**

(Source: Acquired from Google collab)

The “query parser” is used for the extraction of information that is relevant based on certain search criteria.

## Task 2: Document Clustering

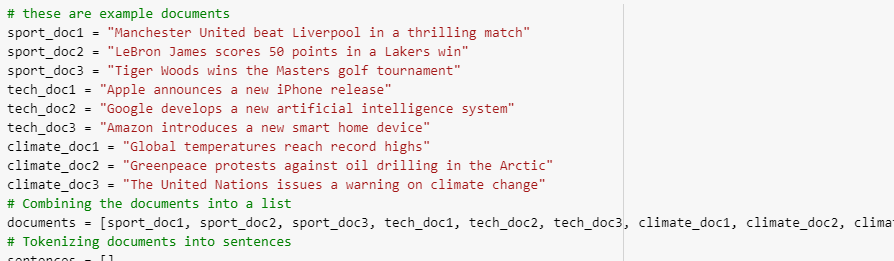
In this section of this project, a system for document clustering is developed. This system is created here with the utilization of the “K-means” clustering along with a proper K value that is determined via the method of error and trial.



**Figure 2.1.1: Importing required libraries**

(Source: Acquired from Google collab)

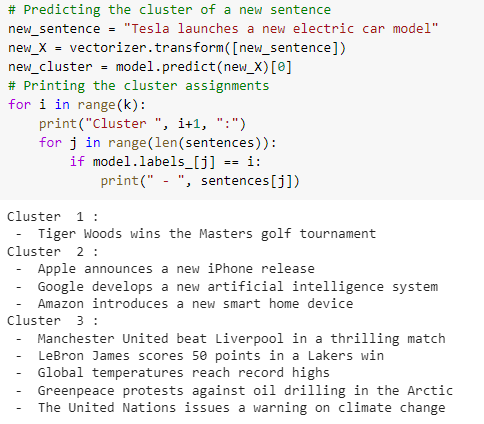
The above figure is the representation of the library that is imported during the execution of the development. The libraries that are imported here are NLTK and K-means.



**Figure 2.1.2: Documents**

(Source: Acquired from Google collab)

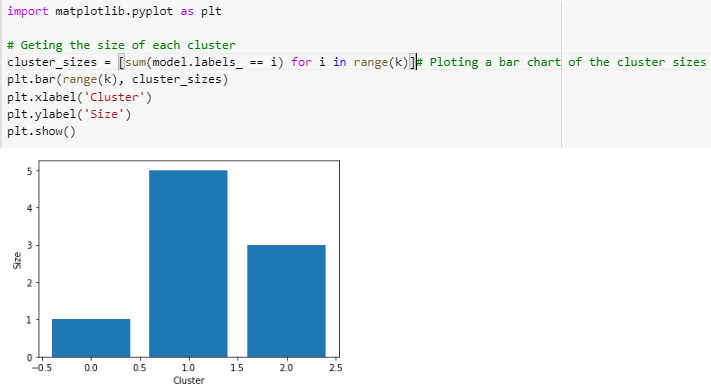
This figure is the visualization of the documents that are provided during the execution of the practical work for developing the system of document clustering with the application of the method “K-means”.



**Figure 2.1.3: Predicting the clusters**

(Source: Acquired from Google collab)

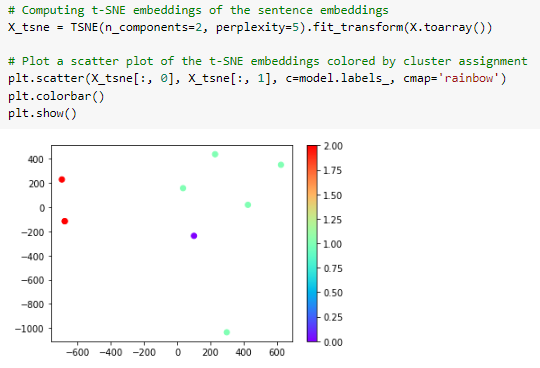
The figure which is attached above is the codes that are written to predict the cluster of a new sentence. After that, the codes to print the clusters are also written to generate the results of the clusters.



**Figure 2.1.4: Bar Plot**

(Source: Acquired from Google collab)

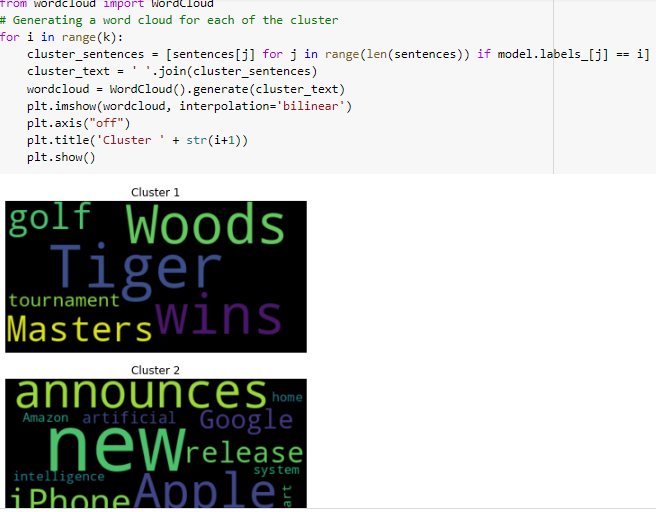
The figure which is presented above is the representation of the bar plot which is generated by the utilization of the clusters that are generated earlier. The size for each of the clusters is provided.



**Figure 2.1.5: Scatter plot**

(Source: Acquired from Google collab)

The above figure is the representation of the scatter plot which is based on the clusters of the documents that are generated above. Before the generation of the scatter plot based on the clusters, the library TSNE is imported from the Python Library.



**Figure 2.1.6: Word cloud**

(Source: Acquired from Google collab)

The figure is the representation of the generation of a word cloud for each of the clusters that were generated from the provided documents. For the generation of the word cloud, a range is provided. The system of document clustering functions is based on the first preprocessing of the documents that include the stemming along with the removal of the stop words. After that, the documents that are pre-processed get represented as a feature vector with the utilization of the technoid like the “bag-of-words model”. After that, the method of “K-means clustering” is applied to the feature vectors which results in the set of cluster generation. Each of the documents is assigned a cluster that contains a centroid that is close to the feature vector. After the completion of the project, it can be stated that all the requirements of this project are fulfilled. In addition to that it can also be stated that this project is done in the consideration of ethical principles.

# Part B (Presentation)

## Introduction

* This project report presents two tasks which are the development of a vertical search engine and the creation of a system for document clustering.
* For the first task, a specialized search engine is going to be developed similar to Google Scholar which is capable to retrieve only the books and papers that are published by CSM.
* The second task of this report includes the collection of a set of documents from multiple categories synch as technology, sport, and many more, and then clustering these utilizing the method of K-means.

In this project report the process of implementation of the principles of informal retrieval is discussed. In addition to that, this report also contains the configuration of the construction of the index along with the definition of the query processor.

## Task 1: Search Engine

### 1.1 Crawler

* This section includes the practical work that is performed to generate the number of staff whose publications get crawled.
* For this task, word tokenizer and stopwords have been imported from the library NLTK which is included in Python.
* After importing the libraries, the codes to fetch the publication link to the page are written.



**Figure 1.1.1: Importing of Word tokenizer**

(Source: Acquired from Google collab)



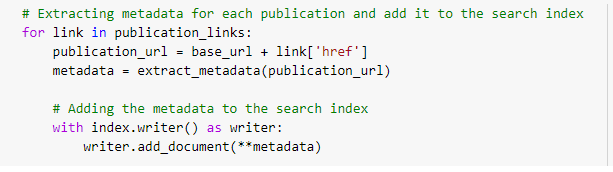
**Figure 1.1.2: Crawling of the search results**

(Source: Acquired from Google collab)

The figures that are attached here are representative of the codes that are written using Python programming language. The crawler gets started by the webpage visiting each of the members of the staff and then retrieving a list that includes their publications.

### 1.2 Indexer

* This section includes the extraction of the classifiers along with the identification of the maximum entropy system.
* To extract metadata for each of the publications the metadata is extracted and then it gets added to the search index.
* The publication URL is defined with the addition of the base URL and link.



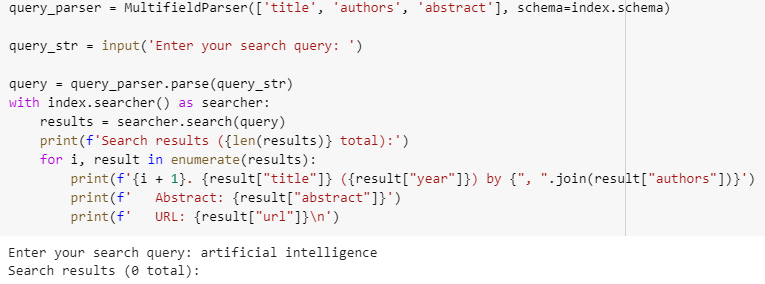
**Figure 1.2.2: Adding metadata to search index**

(Source: Acquired from Google collab)

This figure which is attached here is the representation of the codes that are written to fulfill the requirements. This is done with the utilization of the “Google Colab” song with the application of the Python libraries.

### 1.3 Query Processor

* In this section, at first, the query parser is defined by attaching the title, author, and abstract of the publications.
* After that, an input section is created that requires a search input to perform the desired query.
* Here, ranked retrieval is performed with the utilization of the “Vestor Space model” which can calculate the similarity between each document and the query.



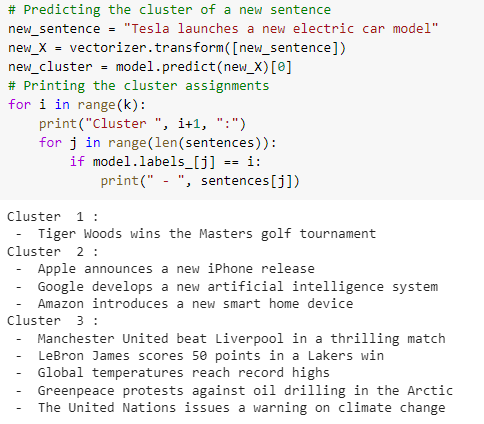
**Figure 1.3.1: Defining query parser**

(Source: Acquired from Google collab)

This section of this project describes the query processor of the search engine that is developed here utilizing the programing language Python. The search engine is able to support both keyword queries and Boolean queries.

## Task 2: Document Clustering

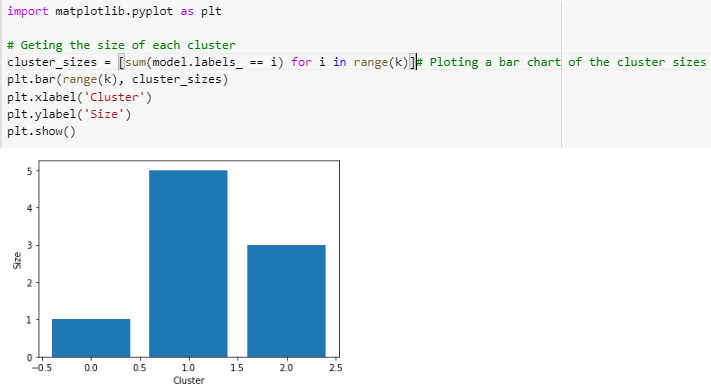
* This section describes the system of document clustering that is developed here.
* The method that is utilized here for the system, of document clustering, is the “K-means clustering” along with a proper K value that is determined via the error and trial method.
* The performance of the clustering algorithm is measured here by the silhouette score which delivers the measurement of the clusters.

)

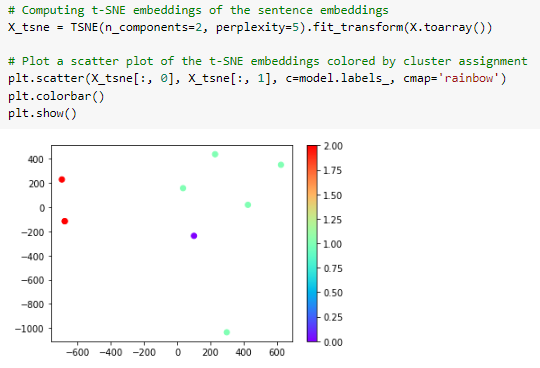
**Figure 2.1.3: Predicting the clusters**

The attached figures are the visualization of the codes that are written to predict the clusters using the “K-means clustering” method.

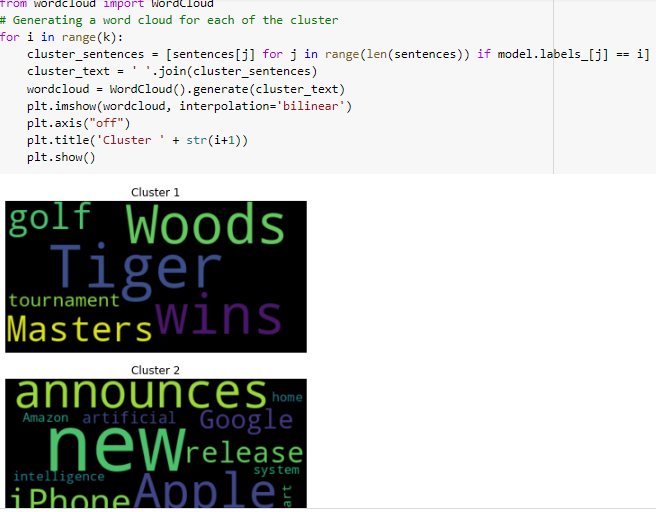
* The robustness and the accuracy of the cluster system are represented here with the attachment of the screenshots.
* This system of document clustering functions by the first pre-processing of the given documents which also includes the stemming and removal of stop words.



**Figure 2.1.4: Bar Plot**



**Figure 2.1.5: Scatter plot**



**Figure 2.1.6: Word cloud**

The attached figure is representing the codes that are written to generate a word cloud for each of the clusters included in the documents that are provided here.

## Conclusions

* In this project work, the development of a search engine along with the development of a system of document clustering is described.
* The search engine is only abler to retrieve the books and papers that are published by CSM at Coventry University.
* The system of document clustering is able to cluster specific documents.

The systems that are developed here are developed with the utilization of proper pre-processing tasks.

# References

Guo, J., Fan, Y., Pang, L., Yang, L., Ai, Q., Zamani, H., ...& Cheng, X. (2020). A deep look into neural ranking models for information retrieval. *Information Processing & Management*, *57*(6), 102067. https://arxiv.org/pdf/1903.06902

Lin, J. (2022, March). A proposed conceptual framework for a representational approach to information retrieval. In *ACM SIGIR Forum* (Vol. 55, No. 2, pp. 1-29). New York, NY, USA: ACM. https://dl.acm.org/doi/fullHtml/10.1145/3576840.3578288

Macdonald, C., &Tonellotto, N. (2020, September). Declarative experimentation in information retrieval using PyTerrier. In Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval (pp. 161-168). https://arxiv.org/pdf/2007.14271

Lin, J., Ma, X., Lin, S. C., Yang, J. H., Pradeep, R., &Nogueira, R. (2021, July). Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 2356-2362). https://dl.acm.org/doi/pdf/10.1145/3404835.3463238

Palotti, J., Scells, H., &Zuccon, G. (2019, July). TrecTools: an open-source python library for information retrieval practitioners involved in TREC-like campaigns. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (pp. 1325-1328). https://ielab.io/publications/pdfs/palotti2019trectools.pdf

Macdonald, C., Tonellotto, N., MacAvaney, S., &Ounis, I. (2021, October). PyTerrier: Declarative experimentation in Python from BM25 to dense retrieval. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (pp. 4526-4533). https://eprints.gla.ac.uk/249268/2/249268.pdf