IR Assignment 5 (Usecase 1)

Aakash Khadka, Gowtham Premkumar, Max Neubauer

Theoretical Background:
In order to solve the assignment we use PyQt5 for the desktop user interface. The searching is done by elasticsearch. The program implements a Model-View-Controller pattern in a three layer architecture (View, Controller, Elasticsearch). Elasticsearch is accessed via API calls on localhost on port 9200. The Application in general is a desktop app to search for Wikipedia topics related to “Computer Science”, “Machine Learning” and “Physics”. After retrieving the most important documents the results are clustered to show them visually using K-means clustering.
This application was developed and tested with Linux.

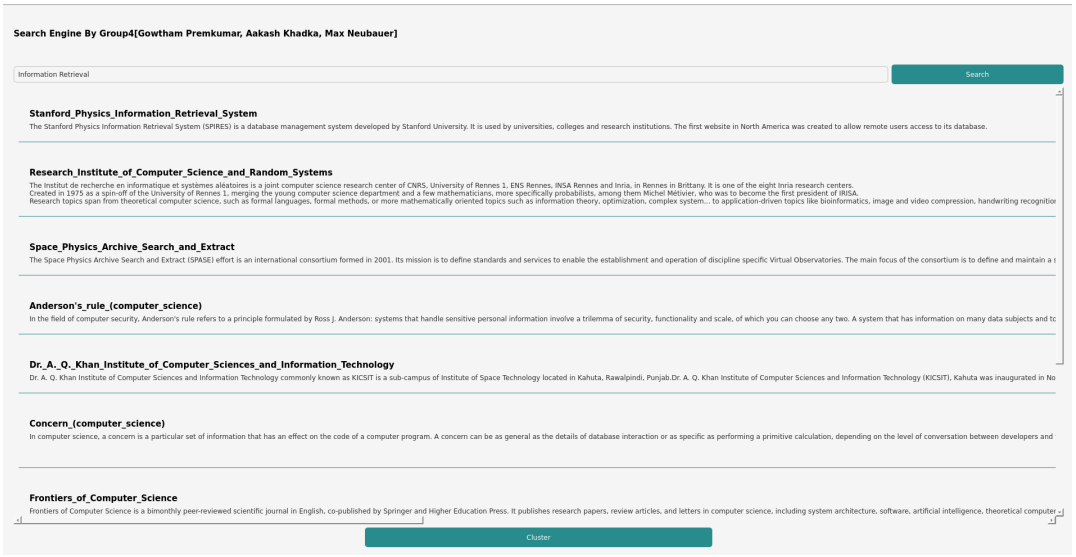
Setup:

- Elasticsearch needs to be installed
- Configure elasticsearch yml file (./elasticsearch-x.x/config/elasticsearch.yml) and adjust according to image 1
- Start elasticsearch (./elasticsearch-x.x/bin/elasticsearch)
- Use firefox to make sure elasticsearch works according to image 2
- Get a subset of all wikipedia topics using download_desired_list_of_topics.py (adjust the path in the wikipedia_topics_path variable)
- With download_new.py the content of the wikipedia topics is downloaded as txt files in the document folder
- The create_index.py creates an empty index with appropriate settings (see Index details below)
- The documents get added to the index by add_documents_to_index.py
- Execute assignment5.py - there will be error messages with missing libraries - install with pip

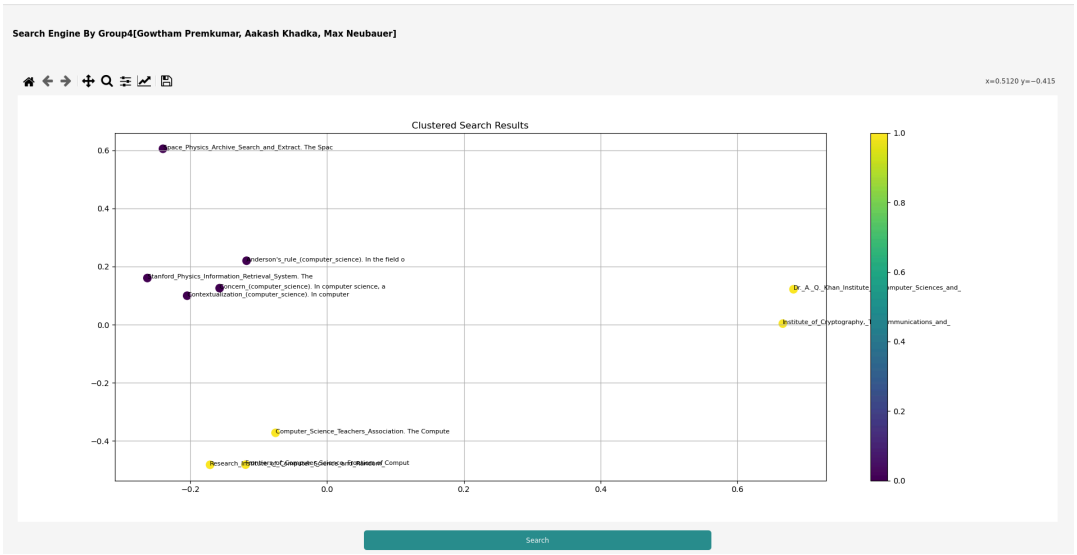
Index details:

- a custom tokenizer is used that tokenizes with “[()-._\s]”
- a custom filter creates n-grams of 1-20 characters
- a custom analyzer uses the custom tokenizer, lowercase-, stop-, and custom filter
- the custom analyzer is applied in the title and the content field of the elastic search documents

Application:



The first image shows the top 10 search results for the query “Information Retrieval”.



The second image shows the clustered view of the top 10 search results for a query. It is always separated into 2 clusters. The results are reduced to two dimensions using Principal Component Analysis.

Appendix:

```
cluster.name: my-application
network.host: localhost
http.port: 9200
xpack.security.enabled: false
```

image 1

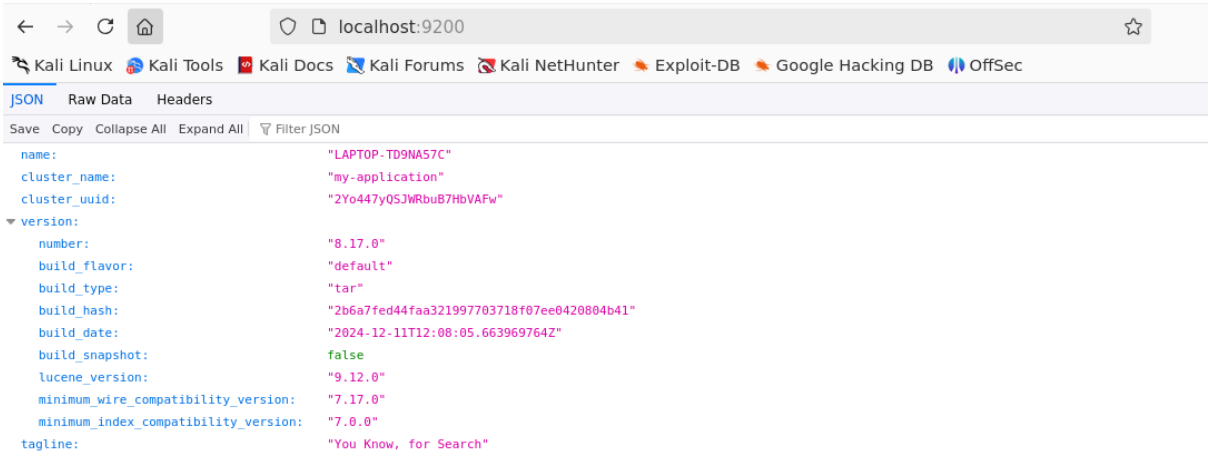


image 2