Information Retrieval (ECS736P) 2022

Assignment 2 – Search Engine Design

Shalini Jain, Benita Ashley, Sushanth Jha, Prajwal B N

Caption based Video Retrieval Search Engine Design

1) Scope

YouTube is one of the most advanced industrial recommendation systems in existence. These recommendations are responsible for helping a billion users discover personalized content from an ever-growing corpus of videos. Due to the high demand and consumption of videos, it is a complex task to find videos based on several attributes which user queries. Most video-based search engines rely on annotations made by the users to identify what the video contains. Relying on these interpretations to perform a query on videos requires an extensive description of the videos and the context which may not always be made available. Thus, we can infer that a video retrieval system that can handle users' queries without the need for such annotations is a relevant topic of research and study.

To overcome this problem, we propose a video retrieval model whose name as it implies can retrieve the videos from the collection that are best described by a particular query by any user. Here, users can input queries not only with video titles but also with several other attributes such as descriptions, captions, and hashtags. For example, "The pursuit of happyness" should return all the relevant videos to the 2006 classic film.

2) Datasets

One of the major tasks in building a search engine is to have the right datasets. We will be using "Data Science YouTube channels video metadata" downloaded from Kaggle which comprises the information of around 60 Data Science YouTube channel video meta-data. We will be evaluating our retrieval model using more than 40,000+ YouTube video meta-data from our dataset.

The dataset document consists of 22 attributes of a YouTube video such as Channel-ID, Title, Video-ID, URL, Video-Title, Description, etc., which we will be using in our search engine to calculate the scores of multiple features of a video and derive the recommendations, based on the user query using a ranking of these feature scores.

3) Architecture

The below diagram illustrates the components that are being used to develop search engine

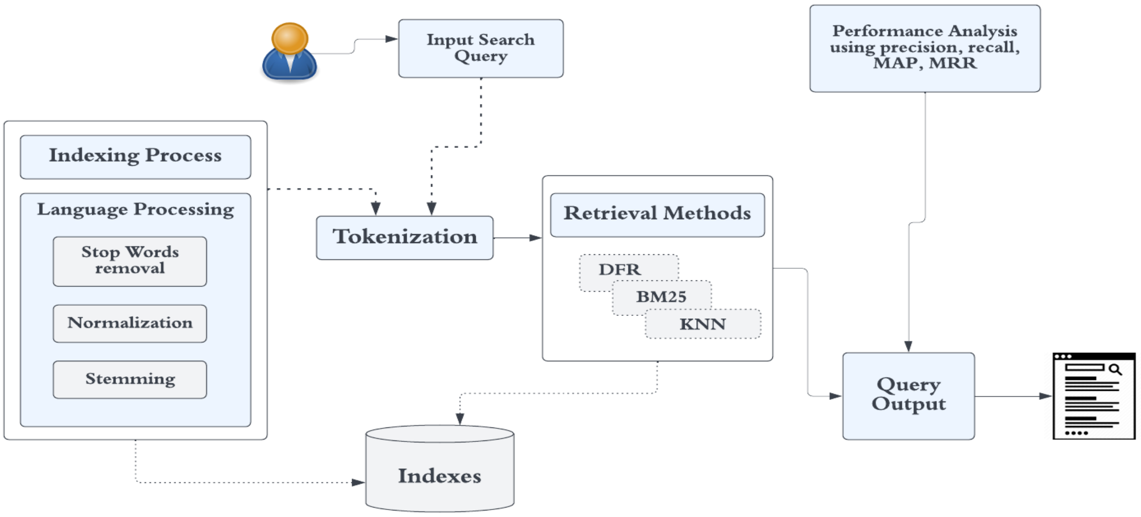


Figure 1 Software Components for search engine

An Information retrieval system consists of three core components i.e Indexing, retrieval, and performance analysis. These components are described as follows:

- **Indexing** is the most important step for efficient retrieval of documents. It further involves four stages as follows:
    - ➢ **Document Collection**: In this project, we are using “Data Science YouTube channels video metadata” for retrieval process
    - ➢ **Tokenization**: This involves the splitting of documents into tokens
    - ➢ **Language Processing:** It involves stop word removal, normalization, and stemming.
    - ➢ **Index Building**: The results of normalization and stemming are used to create an inverted index file. The indexes can also be stored in various other structures for e.g. direct index, document index, etc.
- **Retrieval** methods provide an API to the end-user for querying the application. The model will use the indexes created in the first step for retrieving results.The architecture will implement following algorithms for evaluating the results:
    - ➢ **BM25 (**Best Match 25**)**: Ranking method which is used to estimate the relevance of results based on the search query.
    - ➢ **(Optional) ANN** (Approximate Nearest Neighbor): Used to measure similarity between documents.
- **Query Parser** is used to parse the query and split the query terms using tokenization and pass it to the retrieval process.

- **Performance Analysis** is used to measure the relevance of search results using precision, recall, MAP(Mean Average precision), MRR(Mean Reciprocal Rank).

4) Framework/Tools Employed

We intend to use the Python-Django framework to implement the search engine. This will be integrated with the front-end application for a better user experience. The use of elastic search still needs to be evaluated for better performance.

Tools	Description
Python Django	It will be used to build components necessary for the search engine. Rest API will be exposed to UI to retrieve search results.
Angular	Used to build one UI screen to search and retrieve results
Elastic Search	It will be used to store indexes and also to retrieve search results
Pytest/Unittest	Python framework used to test python code
GitHub	For source control and version management

5) Roles and Responsibilities

An agile delivery approach will be followed by the team in which product features will be developed in an incremental fashion. All members of the team will be assigned data analysis and search engine development activities.

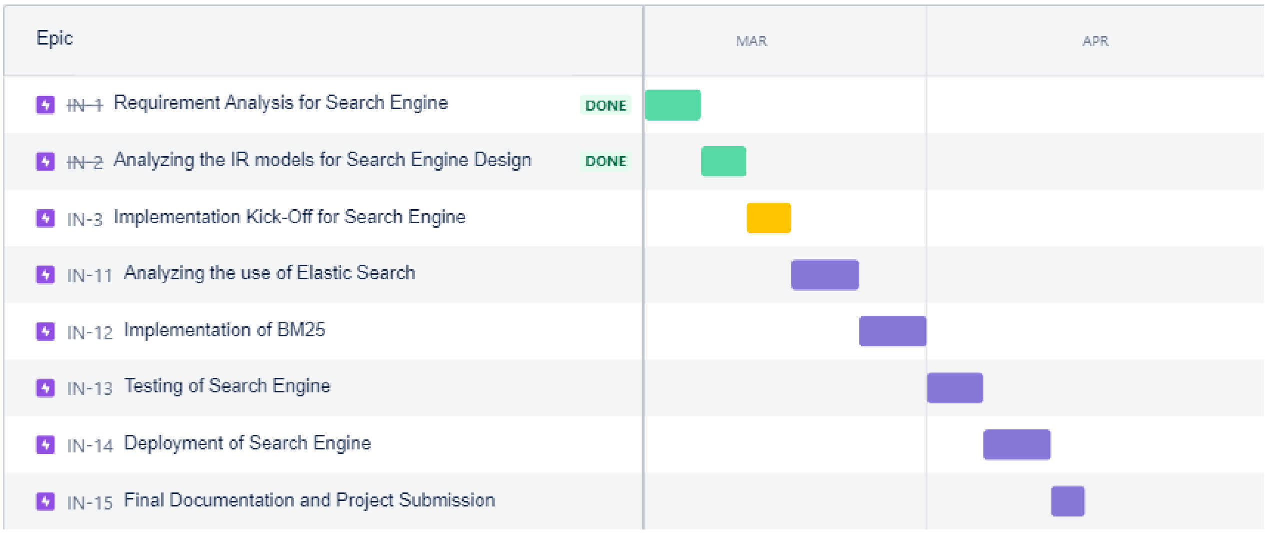
Below are the role descriptions of each team member which are subject to change throughout the search engine development.

Team Member	Roles and Responsibilities
Shalini Jain	Project structure, code setup and analyzing elastic search for better performance. R&D on different methodologies to improve performance in case of multiple features
Benita Ashley	Document creation and implementing indexing and ranking algo.
Sushant Jha	R&D on different methodologies to improve performance in case of multiple features. Implementing indexing and ranking algo
Prajwal B N	Analyzing elastic search, indexing/retrieval framework design, evaluation of results and project planning.

6) Project Timelines

One of the prime factors in delivering a successful project is by adhering to the deadlines and following an effective schedule. Timelines are not only useful in organizing the work, but also it holds everyone accountable for their tasks. Furthermore, They help to set clear direction and priorities by evaluating the deadlines of each task.

We have planned our project in an agile approach with effective timeframes for each phase. We have been tracking all activities of search engine development by following the below time plan which we created during our initial project kick-off meeting. The green bar indicates the completed tasks, whereas the task with yellow timeframe shows the in-progress status and the purple shows the upcoming activities.



Epic	
~~IN-1~~ Requirement Analysis for Search Engine	DONE
~~IN-2~~ Analyzing the IR models for Search Engine Design	DONE
IN-3 Implementation Kick-Off for Search Engine	
IN-11 Analyzing the use of Elastic Search	
IN-12 Implementation of BM25	
IN-13 Testing of Search Engine	
IN-14 Deployment of Search Engine	
IN-15 Final Documentation and Project Submission	

Figure 2 Project Timeplan

7) REFERENCES

1] Simple BM25 extension to Multiple Weighted Fields by Stephen Robertson, Hugo Zaragoza and Michael Taylor Available at http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.9.5255&rep=rep1&type=pdf

2] RankDE: Learning a Ranking Function for Information Retrieval using Differential Evolution by Danushka Bollegala, Nasimul Noman, Hitoshi Iba available at https://danushka.net/papers/danushka_SIGECF_2011.pdf

3] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Infor- mation Retrieval*. Addison-Wesley, 1999

4] Bikas Katwal article on Extended Reciprocal Rank at https://towardsdatascience.com/extended-reciprocal-rank-ranking-evaluation-metric-5929573c778a

5] Information Retrieval systems and Web Search Engines: A Survey by R. Arun Kumar, M. A. Jabbar, Y.V. Bhaskar Reddy available at- https://ijaers.com/uploads/special_issue_files/1502283697-NCTET-CSE-25.pdf

6] Dataset source - Kaggle - https://www.kaggle.com/themlphdstudent/data-science-youtube-video-meta-data