Netflix Film and Series Search Engine

Retrieval scenario:

Content providers, such as Netflix, Amazon Prime, Disney+, etc., maintain vast libraries of content for their users to access. Finding appropriate content to what their users request requires the use of a creative search engine capable of parsing plain text queries, as well as identifying any particular tags in a query, say a request for action films or for films starring a particular actor. Our search engine will use the Netflix database of films and series, and will produce a search engine capable of identifying films from queries, by the extraction of tags and by relevance matching from queries and film descriptions.

Minimal Viable Product (MVP)

The aim is to produce a search engine that can return film or series recommendations from the Netflix dataset, based on an input query. The model will perform feature extraction on the dataset, as well as Stop Word removal, stemming/lemmatization, and tokenization on the film descriptions. The same preprocessing will be performed on the query, and the search results will be based on tags extracted from the query, as well as relevance matching with the film descriptions. An aim is to establish appropriate weightings between the importance of extracted features (tags) and the relevance between the query and the content descriptions.

Enhancements time permitting

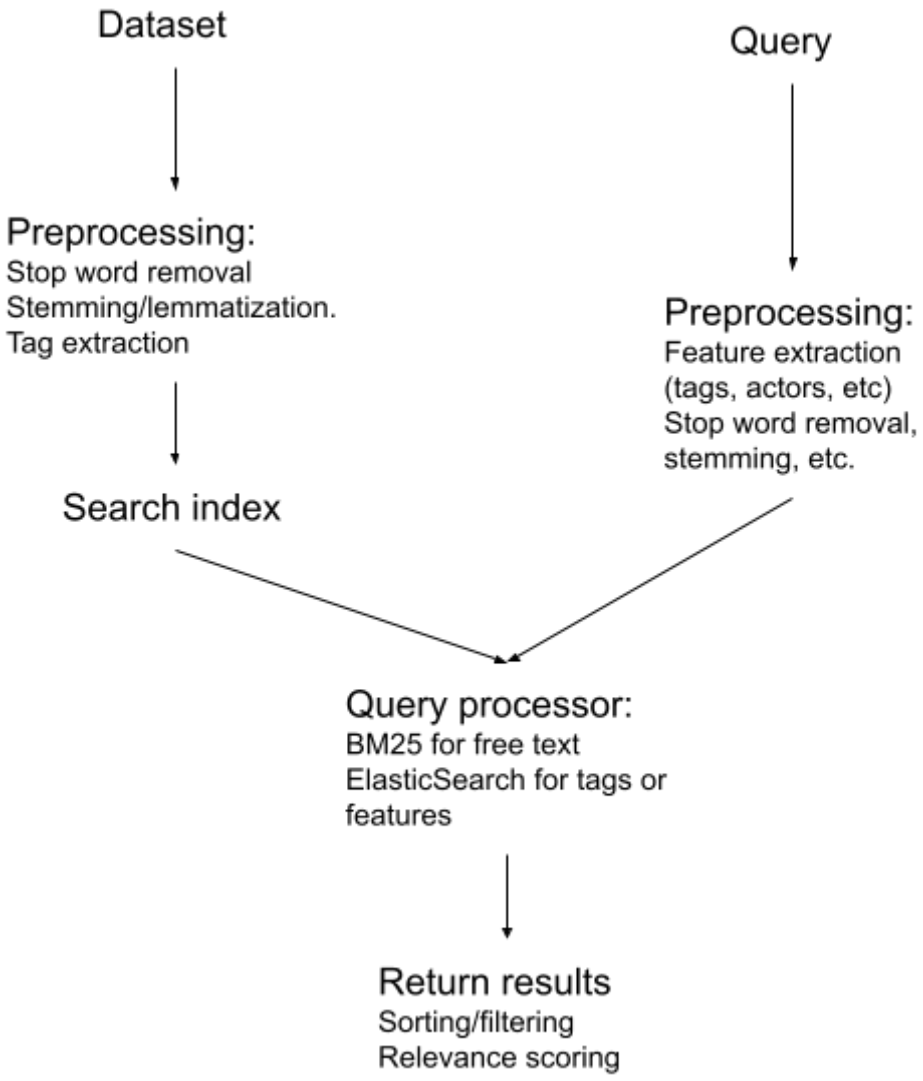
Being able to choose how the results are ordered, like by review scores.
Filter results by various parameters, such as age rating, language, film or series, etc.
Further extension - Include a graphical interface for the search engine.

Data:

Kaggle Netflix dataset
https://www.kaggle.com/ashishgup/netflix-rotten-tomatoes-metacritic-imdb. The dataset provides data on around 15,000 series or films in the Netflix database. Alongside the titles are other features, such as the genre, age rating, etc., along with a short description of the content, which is the part that will be used to perform relevance matching.

Architecture:

The diagram below illustrates the basic structure of the search engine to be developed:



Preprocessing:

Stop word removal, stemming, and lemmatization using spaCy Python library, as well as tag extraction.

Indexing Framework

Preprocessing of content descriptions.
Indexing of tags and processing descriptions into ElasticSearch database.

Retrieval Framework

ElasticSearch for finding relevant results based on tags, with weightings applied to those results.
BM25 to identify relevance between query text and content descriptions.

Query Processor

Preprocessing of the query, as well as the extraction of tags such as content genres, actors, or directors. These tags will have different weightings, based on the ideal retrieval method implemented.
Relevance estimation between queries and films based on the text from the query compared to the description of the content found in the Netflix dataset.

Relevance Analysis

Benchmark retrieval datasets based on a few example queries will be used to evaluate the various weightings and models to identify the best parameters.

Framework:

Tool / API	**Description**
ElasticSearch	Primary search/index API. It has been selected for its ease of use with tags.
BM25	The BM25 ranking function will be used in tandem with ElasticSearch to identify relevance between query text and content descriptions
spaCy	Provisional choice for preprocessing.
GitHub	Source control and development collaboration.

Time management:

Week 1 05 - 12	**Week 2 12-19**	**Week 3 19-23**
Exploratory analysis of the Dataset and familiarisation with the ElasticSearch environment.	Implementation of preprocessing, indexing and retrieval framework, and query processing to meet minimum viable product requirements.	Potential enhancements (time permitting).

Roles/Responsibilities:

All team members will be undertaking development and analysis. The Agile development method (Collier, 2012), will be employed, as it will permit to develop features incrementally, and provide more leeway in the addition of enhancements.

Below are *provisional* role descriptions. These are subject to change.

Team Member	**Responsibilities**
Connor	Preprocessing of queries and dataset for feature and relevant text extraction.
Francesco	Processing of extracted features and text from queries and dataset to provide relevant results.
Mathew	Indexing the dataset for use in the search engine.
Sidthaarth	Return sorted and relevant results to queries.

References:

- Collier, K., 2012. *Agile analytics: A value-driven approach to business intelligence and data warehousing*. Addison-Wesley.
- Elastic.Co, ,2021, *Search APIs*, https://www.elastic.co/guide/en/elasticsearch/reference/current/search.html [accessed 04/07/21]
- Explosion, spaCy, 2021, *Linguistic Features,* https://spacy.io/usage/linguistic-features [accessed 04/07/21]
- Robertson.S, Zaragoza.H, 2009, *The Probabilistic Relevance Framework: BM25 and Beyond,* Foundation and Trends in Information Retrieval, vol 3, no. 4, pp. 333 - 389.