TTDS Group Project

Movie Index - Search Engine for Internet Movie Database

S2322928, S1837774, S2449890, S2402728, S2443454

Mar 2023

# Abstract

# Introduction

Movies, TV shows, anime has become the most common means for people to get relaxed and seek pleasure while enjoy the the leisure time. While the trend of these films and TV shows dominating the large portion of daily entertainment market, it could require some efforts to find the desired films which people wish to watch and the result could be fairly frustrating. Thus, a more advanced search engine targeting on these information can be significantly useful in the real-life scenario.

This project, therefore, was designed to address the problem of struggling in finding films to watch by providing a more progressive solution which embedded a bunch of carefully selected feature.

In comparison to the search function offered by [IMDB.com](http://IMDB.com)*,* which is only able to query the title of films, this search engine dives deeper into the relevant information available for any film such as storyline, genre, keywords to present users more possibilities when they are trying to find a film to enjoy. Essentially, the concept behind this is to explore every single attribute of a film and presenting the result obtained by ranking algorithms, which will be described in later sections. Furthermore, it also provides advanced search to fulfil the real-life scenario needs - such as searching by keywords and genre, filtering the result by colour information, year of releasing, etc.

This system has embedded with nearly 115,000 documents in XML format, being a subset of the Internet Movie Database (IMDB), which was available to be downloaded from https://data.mendeley.com/datasets/rth2kr5hxf/1. Additionally, in order to make the system more scalable in the future, a fully-functional web scraping class was integrated into the system so that more up-to-date data can be effortlessly appended to the existing dataset from time to time.

Regarding the development process, this project was built by a team of 5 students from different disciplines and the strictly followed the concepts of Agile software development life cycle. As the sprint was weekly, all the formal meetings held and meeting minutes file was documented in order to track the progress periodically. While the UML design of the project was unfortunately lost, the mock-up of the original front-end design was provided alongside with the code in the *Design* folder.

Upon the successful completion of the most basic function of the search engine, more advanced features such as auto-translation detection, query expansion, searching suggestion, relevant movie redirection, etc was implemented and will be discussed more in-depth in the section of ***Additional Features***.

# System Design

# Data Collection and Storage

# Preprocessing

# Indexing

# Ranking of Results

# Advanced Search

# API

# GUI

# Evaluation

# Development Life Cycle & Individual Contribution

# Additional Features

# Documentation

# Test

Here, we are using *unittest* to test whether the backend query functionality implementation is successful. We have tested multiple attribute queries, such as *general search, keywords search, title search, language search* and *genres search*, as well as two advanced query methods, *proximity search* and *phrase search*. While the three Boolean queries (AND, OR, NOT) and color filters will be manually tested on the webpage.

We create two test classes, one without the year filter and one with the year filter, the former using the partial dataset from the web crawl of imdb and the latter using the 115,000 documents provided in the original download link for the full test. The test data set without the year filter is smaller, and the query execution time is shorter. Here we randomly choose 2 documents and we set rules to: if we select the document id in the first fifteen results, then as the test passed, the seven test results are passed and are no more than 3 ms. The tests containing the year filter were performed on 7 randomly selected document ids, similar to those described above, except that we did not set a range of results for these two tests to meet the top 15 query ranking requirement due to the less targeted nature of the genres and languages searches. The results took a total of 2990 ms, except for the keywords search, which took 1664.5 ms, but the other tests did not exceed 1s.

660665, Downsyde AND Hilltop Hoods, color, 3.255367 ms

256395, Murder AND giant spigot OR Humanhog, color, 1684.5963 ms.

127365, Danny McGlone NOT father, black and white, 5262.062073 ms

The above are, in order: the selected document id, the query word and Boolean variable entered, the film colour (black and white/colour), and the time taken for the query. The desired result for each query is at the top of the results ranking, with high accuracy. The query times for the web tests are often slower overall than the back-end tests, and the guess here is that it may be that the web pages are taking longer to render.