Task 2.4: Searching with Lucene (practical)

In this exercise, we use Lucene and its fuzzy retrieval model to search for movie titles. You can find the data set with 1000 movie titles on the course page and on Kaggle.com: https://www.kaggle.com/datasets/harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows.

Prioritize functionality and maintain a simple text-based interface. Offer a menu within the program or use arguments for your program to invoke the different sub-tasks. This task is open-ended, and the following sub-tasks are suggestions for what you can achieve with Lucene. Use timeboxing (allocate a specific number of hours) and stop once you encounter difficulties. Collaborate with fellow students to share strategies for tackling the challenges, as navigating through the Lucene documentation can be time consuming.

a) **Import the CSV data into a Lucene index.** Create a Lucene index and implement code to read the CSV file, extracting relevant information (e.g., movie titles, descriptions, ratings) and adding them to the index.

b) **Implement a basic search function to retrieve movies based on keywords.** Create a simple search function that allows users to input keywords and retrieve movies that contain those keywords in their titles or descriptions.

c) **Enhance the search results by considering more relevance factors.** Consider factors like movie ratings, release dates, and occurrences of keywords in different fields to improve the ranking of search results.

d) **Enhance query term matching with query expansion.** Enable fuzzy keyword matches and expand queries if not sufficient search results are provided (or trigger by interface to expand query automatically).

e) **Implement faceted search to allow users to filter results.** Create facets for genres, release years, or other relevant categories to enable users to narrow down search results. Use a feedback prompt to ask users to further filter results

f) **Add spell-checking**. Implement features that correct misspelled queries.

g) **Implement pagination for search results.** Divide search results into pages, allowing users to navigate through multiple pages of results. You can output 10 results and prompt for ‘next’ or ‘prev’ to navigate.