Movie Search Engine

Overview

This project implements an information retrieval system for searching movies using TF-IDF scoring and gamma encoding compression. The system processes movie metadata, builds a compressed inverted index, and enables natural language queries to find relevant movies based on their descriptions and metadata.

Key Features

- Text preprocessing using NLTK and SpaCy for tokenization, lemmatization and stop word removal
- TF-IDF scoring for term weighting and document ranking
- Compressed inverted index using gap and gamma encoding
- Cosine similarity for query-document matching
- Interactive command-line search interface

Technical Implementation

Data Processing Pipeline

1. **Document Loading and Preprocessing**

    - We download the movie dataset from Kaggle, link: https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset?resource=download&select=movies_metadata.csv (https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset?resource=download&select=movies_metadata.csv)
    - Loads movie metadata from CSV file (see `load_movies_csv()`)
    - Creates document representation combining title, overview and genres
    - Performs text preprocessing:
        - Tokenization
        - Stop word removal
        - Lemmatization using SpaCy
        - Punctuation removal

2. **Indexing**

    - Builds inverted index mapping terms to document IDs
    - Compresses postings lists using:
        - Gap encoding to store differences between document IDs
        - Gamma encoding for binary compression
    - Computes TF-IDF scores for all term-document pairs

3. **Search Implementation**
    - Processes search queries using same preprocessing pipeline
    - Computes query vector using TF-IDF weights
    - Ranks documents using cosine similarity
    - Returns top 5 most relevant results

Code Structure

Main Components

- `main.py`: Core implementation of the search engine
- `sri_encodings.py`: Compression utilities for index encoding

Key Functions

- Document processing: `preprocess_document()`, `process_documents()`
- Index building: `build_inverted_index()`, `compute_tfidf()`
- Search: `search()`, `compute_cosine_similarity()`
- Compression: `gap_encode()`, `gamma_encode()`, `compress_gamma_binary()`

Usage

1. Install dependencies:

```
pip install nltk spacy
python -m spacy download en_core_web_sm
```

2. Ensure movie dataset (`movies_metadata.csv`) is in project directory

3. Run the script:

```
python main.py
```