Movie Clips Dataset Creation Methods

Diego Rodrigues

October 15, 2024

Outline

Introduction

Overview

- **Objective:** Develop a comprehensive dataset of movie clips for analysis, visualization, and various applications.
- **Components:**
  - Movie Metadata Collection
  - Library Availability Determination
  - Plot Embedding Generation
  - Data Cleaning and Preparation
  - Embedding Visualization

Dataset Description

Dataset Overview

- **Source:** `sample_mflix.embedded_movies` from Hugging Face.
- **Attributes Collected:**
  - `_id`: Unique identifier for the movie.
  - `title`: Title of the movie.
  - `release year`: Year the movie was released.
  - `genres`: List of genres (e.g., Western, Action, Fantasy).
  - `plot`: Brief summary of the movie's plot.
  - `plot_embedding`: Numerical embeddings of the plot generated using OpenAI's `text-embedding-ada-002` model.
  - `runtime`, `rated`, `cast`, `directors`, `writers`, `awards`, `imdb`, `countries`, `tomatoes`, etc.
- **Size:** 1,500 movie records.
- **Format:** JSON documents.

Data Acquisition Methods

- **Hugging Face Dataset:**
  - Utilized the `sample_mflix.embedded_movies` dataset.
  - Contains movies with genres: Western, Action, or Fantasy.
- **APIs Used:**
  - IMDb API for additional movie metadata.
  - TMDb API for supplementary details and multimedia resources.
- **Data Extraction:**
  - Automated scripts to fetch and integrate data from multiple sources.
  - Ensured compliance with data usage policies of respective platforms.
- **Batch Processing:**
  - Employed threading and asynchronous requests to enhance efficiency.
  - Managed large-scale data extraction seamlessly.

Determining Library Availability

Library Availability Process

- **Objective:** Identify which movies are available in partnered libraries.
- **Method:** Utilize the Primo API to check availability.
- **Steps:**
  1. Extract movie titles from the dataset.
  2. Query the Primo API for each title.
  3. Parse API responses to determine availability.
  4. Attach library links to available movies.
- **Output:** Enhanced dataset with library availability status and links.

API Integration for Availability

- **Primo API Usage:**
  - Fetch availability status for each movie.
  - Retrieve detailed library links where the movie is available.
- **Concurrency:** Implemented multithreading to handle multiple API requests simultaneously.
- **Error Handling:** Managed API rate limits, connection issues, and unexpected responses.
- **Data Storage:** Stored results in JSON and Excel formats for flexibility.

Data Processing and Embedding

Handling Missing `plot_embedding`

- **Issue:** Some movie records lack valid `plot_embedding`, leading to processing and visualization errors.
- **Strategy:**
  - **Identification:** Detect records with missing or empty `plot_embedding`.
  - **Exclusion:** Exclude these records from further processing to maintain data integrity.
  - **Logging:** Log the titles of excluded movies for future reference or manual inspection.
- **Benefits:**
  - Prevents errors during library availability checks and visualization.
  - Maintains the quality and reliability of the dataset.

Plot Embedding Generation

- **Purpose:** Convert textual plot summaries into numerical vectors for analysis.
- **Techniques Used:**
  - **OpenAI's** `text-embedding-ada-002`**:** Generates high-dimensional embeddings capturing plot semantics.
- **Tools and Libraries:**
  - Hugging Face Transformers, OpenAI API.
- **Output:** 1536-dimensional plot embeddings for each movie.

Data Cleaning and Preparation

- **Handling Missing Values:**
  - Ensured completeness of essential fields (title, plot).
  - Excluded records with critical missing data (`plot_embedding`).
- **Normalization:**
  - Standardized numerical features (ratings, runtime).
  - Encoded categorical variables (genre, language) using one-hot encoding.
- **Data Validation:**
  - Verified consistency and accuracy of data entries.
  - Removed duplicates and corrected inconsistencies.

Embedding Visualization

Embedding Visualization Overview

- **Objective:** Visualize movie embeddings to identify patterns and clusters.
- **Dimensionality Reduction Techniques:**
  - UMAP (Uniform Manifold Approximation and Projection)
- **Visualization Tools:**
  - Matplotlib, Seaborn for static plots.
  - Plotly for interactive visualizations.
- **Insights Gained:**
  - Identification of genre clusters.
  - Trends based on release year and ratings.

Dimensionality Reduction Techniques

- **UMAP:**
  - Maintains both local and global data structure.
  - Faster and scalable compared to other techniques like t-SNE.
  - Ideal for large datasets and preserving meaningful relationships.

UMAP Visualization

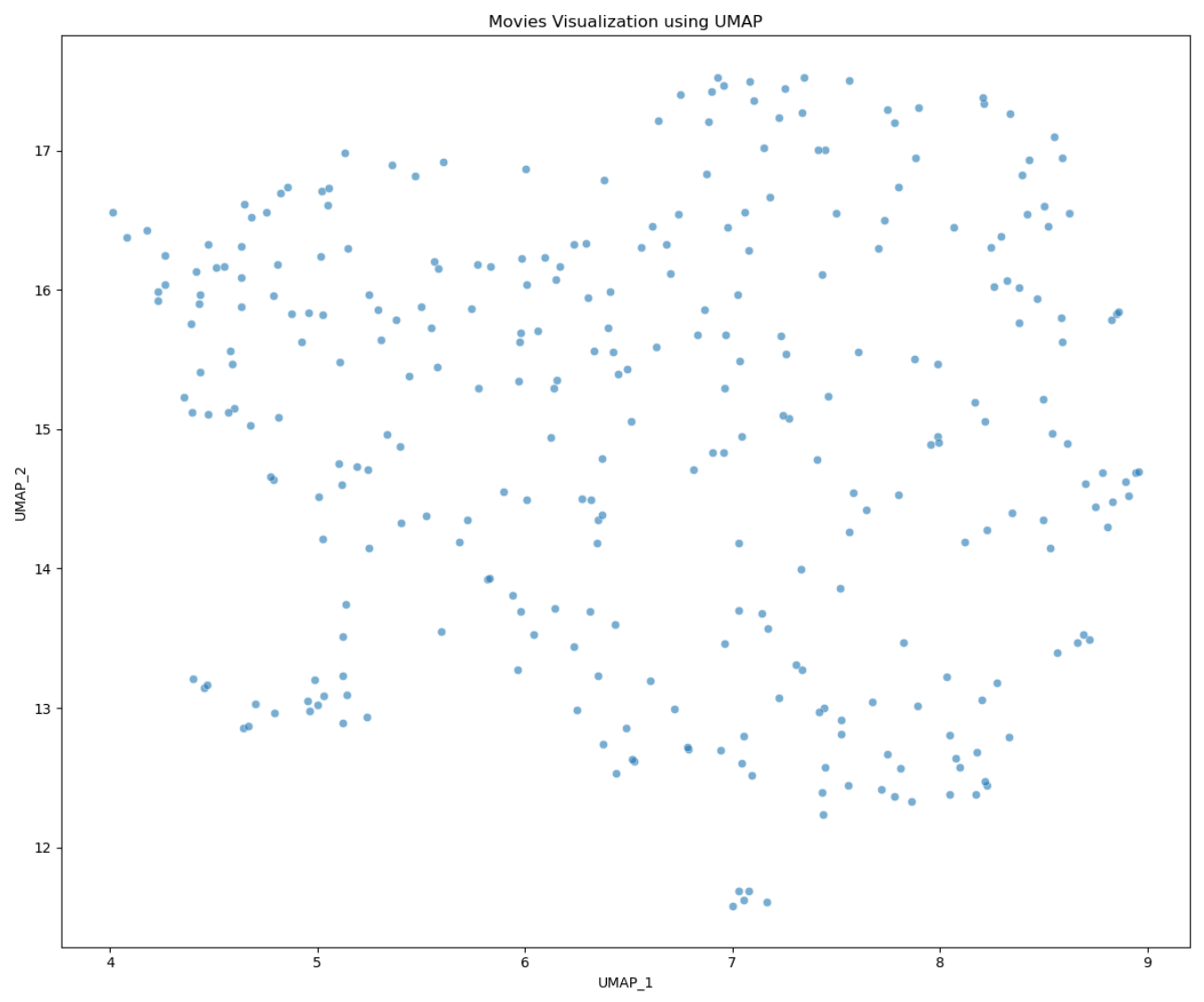

Figure 1: UMAP Scatter Plot of Movie Embeddings

Results and Insights

Dataset Statistics

- **Total Movies:** 1,500
- **Genres:** Western, Action, Fantasy
- **Average Runtime:** 106 minutes
- **Language Distribution:** Primarily English with select multilingual entries.
- **Library Availability:** 60% of movies available in at least one library

Clustering Insights

- **Genre Clusters:** Clear separation between Western, Action, and Fantasy movies in the embedding space.
- **Trend Analysis:**
  - Release year trends showing the evolution of genres over time.
  - Correlation between IMDb ratings and embedding distances, indicating viewer preferences.
- **Library Availability Patterns:**
  - Popular and highly-rated movies are more likely to be available in libraries.
  - Niche genres have lower availability rates, highlighting areas for library collection expansion.