Plot-based Recommendation System Documentation

This documentation provides an overview of the code for a movie recommendation system based on plot similarity using FastAPI, SBERT (Sentence-BERT), and other data processing libraries.

Code Structure

The code is structured into the following sections:

1. Import Statements:
    - The code begins by importing essential Python libraries and modules.
2. FastAPI Initialization:
    - An instance of the FastAPI framework is created and named `plot_app`.
3. SBERT Model Loading:
    - The SentenceTransformer model ("all-mpnet-base-v2") is loaded to generate SBERT embeddings for movie overviews.
4. Data Processing Functions:
    - A function to calculate SBERT embeddings from text, as well as a function to process lists of items, is defined.
5. Movie Data Loading and Cleaning:
    - Movie data is loaded from CSV files ("movies_metadata.csv" and "credits.csv").
    - Data is filtered, cleaned, and limited to the top 1000 most popular movies.
6. Device Configuration:

- The code determines the computing device (CPU or CUDA) for SBERT model processing.

7. Feature Extraction and Embeddings:
   - Genres, overviews, countries, and cast information is processed.
   - TF-IDF Vectorization is applied for genres, countries, and cast.
   - SBERT embeddings are computed for movie overviews.
8. Movie Recommendation Function:
   - A function is defined to recommend movies based on user history (provided as a list of movie titles).
   - Similarity scores are calculated using a combination of feature embeddings.
   - The top 10 recommendations are selected.
9. Sample User History Data:
   - Sample user history data is provided as a list of movie titles.
10. Recommending Movies Endpoint:
    - An endpoint (`/recommend-movies-plot`) is created to recommend movies based on user history.
    - The top 10 recommendations, including movie titles, IMDb IDs, overviews, similarities, and genres, are returned in JSON format.
11. SBERT Embeddings Creation:
    - SBERT embeddings for movie overviews are created as a tensor.
12. Movie Search Function:
    - A function is defined to search for movies based on plot similarity.
    - The similarity scores between the user's input and movie overviews are calculated.
    - The top 10 search results are selected.
13. Movie Search Endpoint:
    - An endpoint (`/search-plot/`) is created to search for movies based on plot

similarity.
    - The user's plot input is processed, and the top 10 search results are returned in JSON format, including movie titles, IMDb IDs, overviews, similarities, and genres.

Usage

1. `/recommend-movies-plot` Endpoint:
    - Access this endpoint via a GET request to receive movie recommendations based on user history.
    - The recommendations include movie titles, IMDb IDs, overviews, similarities, and genres.
2. `/search-plot/` Endpoint:
    - Access this endpoint via a POST request, providing a plot input.
    - The system performs a plot-based search and returns the top 10 search results, including movie titles, IMDb IDs, overviews, similarities, and genres.

Technical Details

- The system utilizes SBERT embeddings to analyze movie overviews and recommend movies.
- Cosine similarity is used to measure similarity between movie features.
- A combination of features, including overviews, titles, locations, cast, and genres, is considered for recommendations.

Acknowledgments

This code provides a foundation for a plot-based movie recommendation system. It can be expanded and improved with a larger dataset, user feedback, and additional user interaction features for a more sophisticated recommendation system.