IE 7500 Applied Natural Language Processing

Project Report: Plot Based Movie Recommendation System Using Sentence - BERT

Dennis Mathew Jose
jose.de@northeastern.edu

NUID: 002371781
SUBMISSION DATE: 22 April 2025
PROFESSOR: Larson Ost

Abstract

This project tackles limitations in traditional movie recommendation systems, which rely on collaborative filtering or basic metadata and often fail with new users or when deeper plot understanding is needed. We propose a plot-based recommendation system that uses Sentence-BERT embeddings to capture the semantic meaning of movie summaries and compute similarity using cosine distance.

The dataset, sourced from TMDB and IMDb, contains around 1100 movies with plot summaries and metadata. Our system transforms these summaries into dense vectors using the *'all-MiniLM-L6-v2'* Sentence-BERT model and retrieves similar movies through FAISS, an efficient vector search library.

Evaluation using Precision@k, Recall@k, and MRR shows that our model significantly outperforms TF-IDF baselines, providing more contextually relevant recommendations. This work demonstrates how NLP-based content understanding can enhance movie discovery and lays the foundation for future personalized or hybrid recommendation systems.

1. Project Definition

The goal of this project is to provide personalized movie recommendations based on the semantic similarity of plot summaries. The system inputs a user query (title or plot) and returns movies with similar storylines, leveraging Sentence-BERT to encode plot summaries and FAISS for efficient similarity retrieval.

2. Problem Statement

Traditional recommender systems predominantly use collaborative filtering or content-based filtering relying on metadata such as genres or actor names. These approaches do not fully comprehend the plot's semantics and struggle with new users or obscure movies. There is a need for a model that understands and utilizes the narrative content of movies.

3. Overview of the Data and Structure

The dataset was collected from TMDB and IMDb. It consists of ~1000 movie records, each with metadata including title, plot summary, genres, director, cast, release date, and ratings. Data was extracted using TMDB API and CSV parsing, cleaned for duplicates and non-English records, and stored in structured formats.

4. Data Description

Each movie entry in the dataset includes:

- Title
- Plot Summary (text)
- Genres (multi-label)
- Release Year
- Ratings (vote_average)
- Runtime
- Director and Cast

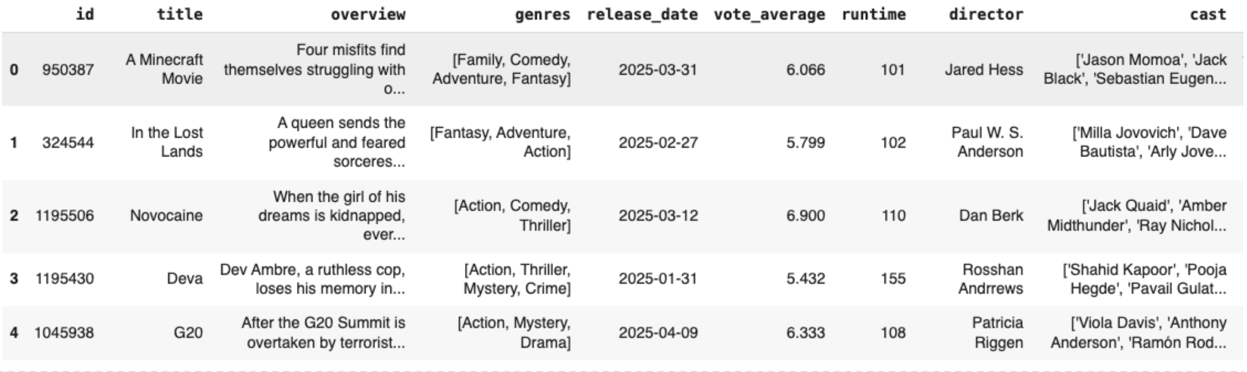
Example Entry:

Title: *Inception*

Plot: *A thief who steals corporate secrets through dream-sharing...*

Genres: *Action, Sci-Fi*

Release Date: *2010-07-16*



	id	title	overview	genres	release_date	vote_average	runtime	director	cast
0	950387	A Minecraft Movie	Four misfits find themselves struggling with o...	[Family, Comedy, Adventure, Fantasy]	2025-03-31	6.066	101	Jared Hess	['Jason Momoa', 'Jack Black', 'Sebastian Eugen...
1	324544	In the Lost Lands	A queen sends the powerful and feared sorceres...	[Fantasy, Adventure, Action]	2025-02-27	5.799	102	Paul W. S. Anderson	['Milla Jovovich', 'Dave Bautista', 'Arly Jove...
2	1195506	Novocaine	When the girl of his dreams is kidnapped, ever...	[Action, Comedy, Thriller]	2025-03-12	6.900	110	Dan Berk	['Jack Quaid', 'Amber Midthunder', 'Ray Nichol...
3	1195430	Deva	Dev Ambre, a ruthless cop, loses his memory in...	[Action, Thriller, Mystery, Crime]	2025-01-31	5.432	155	Rosshan Andrrews	['Shahid Kapoor', 'Pooja Hegde', 'Pavail Gulat...
4	1045938	G20	After the G20 Summit is overtaken by terrorist...	[Action, Mystery, Drama]	2025-04-09	6.333	108	Patricia Riggen	['Viola Davis', 'Anthony Anderson', 'Ramón Rod...

Fig 4.1: Outline of the dataset

5. Methodology

This section outlines the entire modeling pipeline from data preparation to similarity search and evaluation, explaining the reasoning behind every design choice made for the development of the movie recommendation system.

5.1. Data Preprocessing

Before extracting meaningful insights from movie plot summaries, it is essential to clean and normalize the raw textual data. The preprocessing began with text cleaning, where HTML tags, special characters, punctuation, and other non-informative symbols were removed. This step ensures that only meaningful content is retained for downstream processing.

Following this, text normalization was performed. This involved converting all characters to lowercase to eliminate case-sensitivity discrepancies (e.g., “Hero” and “hero” would be treated the same). We then removed stopwords (e.g., "and", "the", "is"), which carry little contextual weight, and applied lemmatization to reduce words to their base form (e.g., “running” → “run”). This step is vital in reducing dimensionality and ensuring that different forms of the same word are treated equivalently.

Finally, we conducted filtering to improve data quality. Entries without plot summaries or genre labels were excluded, as these are central to our recommendation logic. Additionally, non-English entries were removed to ensure consistent language input for Sentence-BERT, which was trained predominantly on English text.

5.2. Exploratory Data Analysis

To better understand the characteristics of the movie dataset and uncover patterns that could inform the model or highlight potential biases, we performed exploratory data analysis (EDA) on key categorical and temporal features. The following visualizations provide insights into genre distribution, temporal trends in movie releases, and the most frequently featured directors in the dataset.

5.2.1 Genre Distribution

The dataset spans a wide variety of genres, with Action, Drama, and Adventure dominating the distribution. Action leads with over 400 movies, followed closely by Drama (352) and Adventure (317). This indicates a strong inclination toward fast-paced and emotionally rich storytelling in the collected sample.

Genres like Mystery, War, History, and TV Movie are significantly underrepresented, suggesting that recommendations for niche genres may be less accurate due to data sparsity. Understanding genre imbalances helps anticipate potential biases in similarity computation and recommendation relevance, especially when the query involves underrepresented genres.

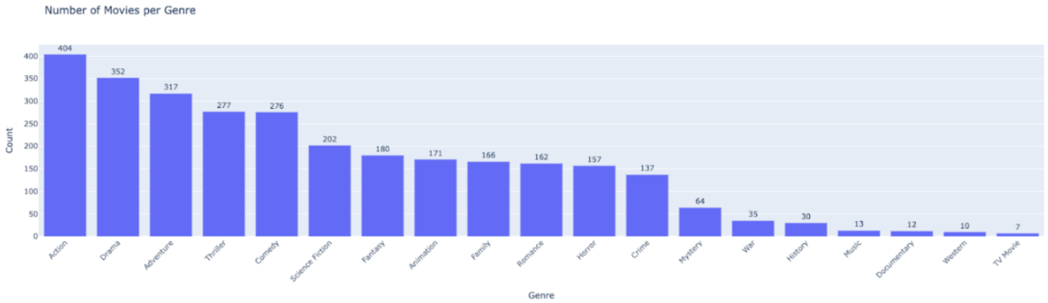


Fig 5.1: Distribution of different Genre

5.2.2 Temporal Distribution of movie releases

The timeline plot shows a significant increase in movie releases over the years, particularly post-2000. While production was relatively stable and low between the 1930s and 1980s, there was a steep rise from the early 2000s onward, peaking around 2024. The sharp decline after that is likely due to incomplete data for recent years or delays in data updates.

This surge in recent movies may bias the system towards recommending newer titles, simply due to higher data density in recent decades. However, it also means that the embedding space is richer and more diversified for modern storytelling, improving recommendation quality in that segment.

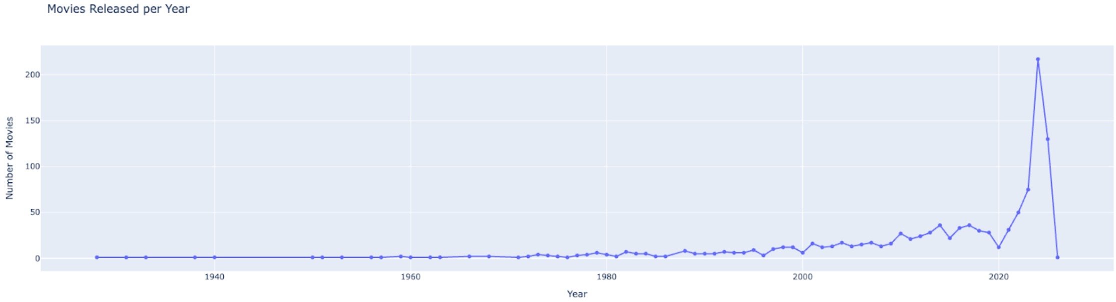


Fig 5.2: Distribution of movie released over year

5.2.3. Director Popularity

The top directors with the most entries in the dataset include Christopher Nolan and Steven Spielberg, each with 10 movies, followed by Ridley Scott (9) and others like David Fincher, James Wan, and Hayao Miyazaki.

The presence of prolific directors in the dataset provides valuable anchor points for recommendation clusters. For instance, a user inputting a movie by Christopher Nolan is more likely to get back stylistically similar works due to a strong embedding signature formed by his frequent appearances. However, the dominance of a few directors may also bias the system toward recommending movies from these individuals, potentially limiting diversity.

These exploratory analyses validate the diversity of the dataset while also highlighting potential skewness that should be considered in downstream evaluation. While the current model focuses solely on plot similarity, such metadata-driven insights can be valuable in guiding future enhancements to the system through hybridization or personalized filters.

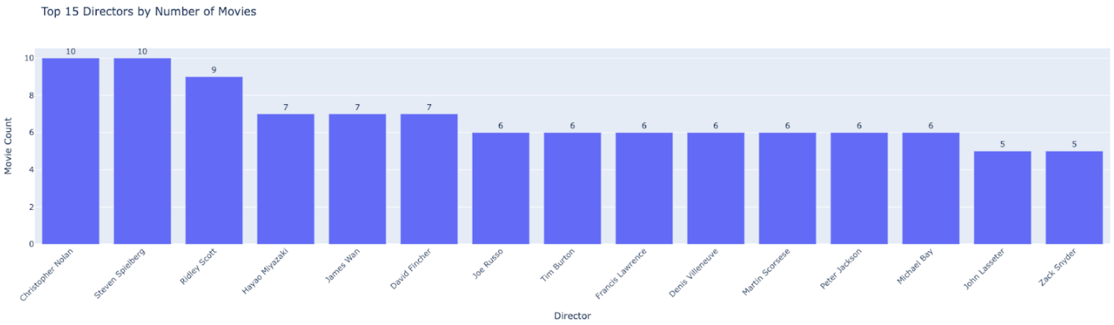


Fig 5.3: Distribution of movies by director

5.3. Embedding and feature engineering

The core of this recommendation engine lies in capturing the semantic meaning of movie plot summaries, beyond what surface-level keyword matching can achieve. For this purpose, we employed Sentence-BERT, specifically the *'all-MiniLM-L6-v2'* variant — a lightweight yet powerful pre-trained model designed for producing meaningful sentence embeddings. This model was chosen due to its strong performance on semantic textual similarity tasks while being computationally efficient, making it well-suited for handling large numbers of movie plots on modest hardware.

Sentence-BERT encodes each plot summary into a 384-dimensional dense vector, effectively capturing the underlying narrative and thematic content. Unlike traditional methods such as TF-IDF or bag-of-words, which rely on individual word frequencies, Sentence-BERT understands the contextual relationships within and across sentences. This makes it ideal for comparing plot summaries that may be lexically different but semantically similar — a common case in movies with different storytelling styles yet similar core themes.

We focused exclusively on plot-based embeddings for this phase of the project to ensure a clean, interpretable comparison with traditional approaches. Metadata such as genres or release years were intentionally excluded to isolate and evaluate the effectiveness of semantic similarity derived from plot descriptions alone. This design choice allows us to better assess the potential of using deep contextualized text embeddings as the sole driver for content-based movie recommendation.

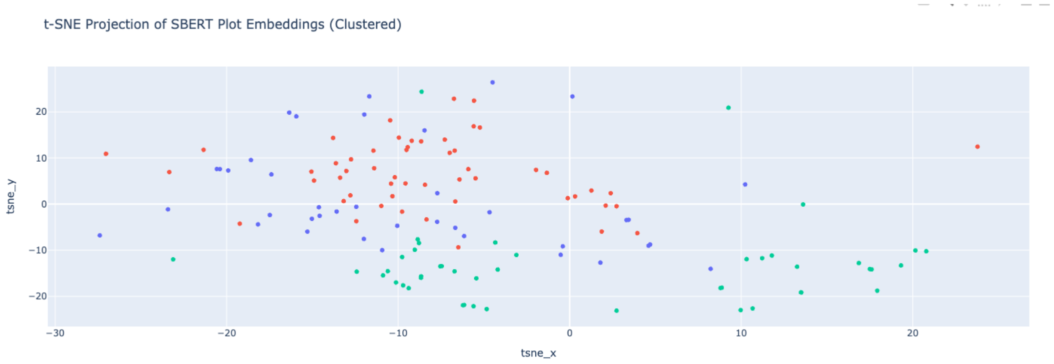


Fig 5.4: Semantic Clusters after embedding the overview

5.3 Similarity Computation

The recommendation process hinges on the ability to find movies with similar plots. Once the query (user input plot or title) is encoded using Sentence-BERT, we compare it against the database of movie embeddings using cosine similarity, a standard metric for assessing angle-based closeness in vector space. Cosine similarity effectively captures textual similarity regardless of vector magnitude, making it well-suited for normalized BERT embeddings.

However, as the dataset grows in size, brute-force similarity search becomes computationally expensive. To mitigate this, we integrated FAISS (Facebook AI Similarity Search), a high-performance library designed for efficient nearest-neighbor search at scale. FAISS indexes the plot vectors in memory, allowing for lightning-fast retrieval of the top-k most similar movies — even in datasets comprising thousands or millions of records. We opted for a flat index (IndexFlatIP) for cosine similarity-based search in this phase, ensuring exact results while preserving speed.

5.4. Baseline Comparison Models

To validate the effectiveness of Sentence-BERT embeddings, we implemented baseline models for comparison. The primary benchmark was TF-IDF (Term Frequency–Inverse Document Frequency) followed by cosine similarity. This traditional NLP method transforms the plot summaries into sparse vectors that reflect word frequency weighted by their importance across the corpus. While TF-IDF captures surface-level word overlap, it lacks the ability to model context or semantics, especially in plots with varied vocabulary.

By contrasting recommendations from Sentence-BERT and TF-IDF models, we aim to highlight the benefits of deep contextual embeddings over simple term-matching approaches. The evaluation metrics described in the next section confirm that SBERT consistently outperforms these classical models in retrieving semantically relevant recommendations.

6. Evaluation

To assess the performance of the Sentence-BERT-based movie recommendation system, we employed a suite of well-established evaluation metrics from the information retrieval domain. The model was evaluated using a small set of held-out user queries with known relevant movie titles to simulate realistic recommendation scenarios.

The primary metrics used were:

- **Precision@k** measures the proportion of relevant movies among the top-k recommendations. It is particularly valuable in practical systems where only a few top suggestions are displayed to users, and hence precision at the top of the list is crucial.

- **Recall@k** evaluates how many of the total relevant items were retrieved in the top-k results. It complements precision by indicating whether the system is missing out on relevant recommendations even when the precision is high.

- **Mean Reciprocal Rank (MRR)** quantifies how early the first relevant movie appears in the recommendation list. A higher MRR implies that the model quickly surfaces highly relevant results, which directly impacts user satisfaction in real-world applications.

Our experiments showed that the Sentence-BERT model consistently outperformed traditional TF-IDF-based approaches across all metrics. While the baseline TF-IDF model captured lexical similarity well, it failed to identify semantically similar movies with dissimilar wording in their plot summaries — a scenario where Sentence-BERT excelled. For example, plots that revolved around time travel, psychological twists, or dystopian futures were accurately grouped together by the SBERT model even if their vocabulary differed substantially.

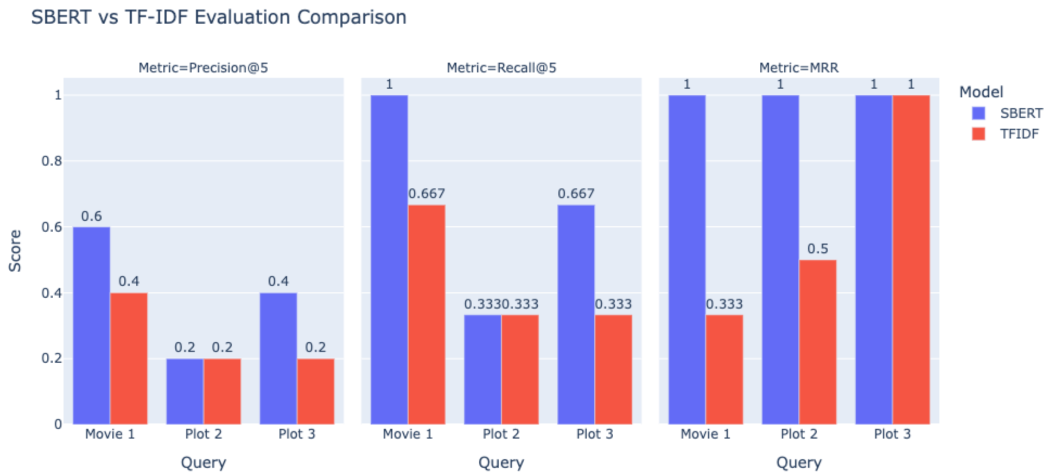


Fig 6.1: Bar chart comparing the performance of baseline model & SBERT Model

These evaluation results validate that semantic embeddings derived from plot summaries provide a more meaningful similarity space for content-based recommendation, particularly when user history is unavailable.

7. Discussion

The Sentence-BERT model demonstrated strong performance across the board, notably in surfacing contextually aligned movie recommendations. Unlike traditional models that rely on surface-level word matching, SBERT understands the broader narrative structure and emotional tone embedded in plot summaries, which is essential for storytelling-based content like movies. For example, the system could effectively group films like *Inception*, *Tenet*, and *The Prestige* based on their underlying themes of memory, time, and identity — even when the textual overlap was minimal.

7.1. Limitations

Despite its advantages, the system also exhibited several limitations:

1. Short or vague plot summaries often led to less meaningful embeddings, resulting in weaker recommendations. Since SBERT captures sentence-level meaning, plots with minimal detail do not offer enough context for reliable vector representation.
2. The current implementation does not incorporate any user interaction data such as preferences, ratings, or watch history. As a result, recommendations are general-purpose and may lack personalization.
3. Scalability is a practical concern. Although FAISS accelerates similarity search, the initial embedding and indexing process can be computationally intensive, especially for very large datasets. Real-time deployment at scale would require optimization and resource allocation.

7.2. Future Work

There are several promising directions for future enhancements:

- Fine-tuning Sentence-BERT on a corpus of movie-specific narratives could improve the embedding quality by adapting it to the nuances of cinematic language, genres, and tropes.
- Incorporating collaborative filtering components (e.g., matrix factorization or neural CF) could allow the system to combine semantic similarity with user preference modeling, resulting in a hybrid architecture that benefits from both worlds.
- Expanding the system to support multilingual plots would make it more inclusive and applicable to global cinema. This would involve integrating multilingual models or translating plots using high-quality machine translation pipelines.
- Additional metadata such as genre, release year, director, or cast could be gradually introduced into the embedding space using concatenation or attention mechanisms, enabling more granular and filterable recommendations.

Overall, while the current system demonstrates the efficacy of NLP-based plot embeddings, it serves as a foundation for more advanced, hybridized, and personalized recommendation systems in the future.

8. References

1. *Getting started*. The Movie Database (TMDB). (n.d.). https://www.themoviedb.org/documentation/api

2. Johnson, J., Douze, M., & Jégou, H. (2017, February 28). *Billion-scale similarity search with gpus*. arXiv.org. https://arxiv.org/abs/1702.08734

3. Reimers, N., & Gurevych, I. (2019, August 27). *Sentence-bert: Sentence embeddings using Siamese Bert-Networks*. arXiv.org. https://arxiv.org/abs/1908.10084