UIUCinema: Project Report

Yichong Liu
University of Illinois
Urbana-Champaign
Champaign, IL, USA
yl200@illinois.edu

Theodore Xiong
University of Illinois
Urbana-Champaign
Champaign, IL, USA
tyxiong2@illinois.edu

Mei Han
University of Illinois
Urbana-Champaign
Champaign, IL, USA
meihan3@illinois.edu

Abstract

We have all experienced the frustration of remembering a movie's plot but forgetting its title. Most search engines fail in this scenario because users rarely describe a story using the exact keywords found in a database. To address this, we developed **UIUCinema**, a search engine designed for plot-based retrieval. We propose a Hybrid Search Model that combines traditional keyword matching (**BM25**) with semantic understanding (**Sentence Transformers**) to bridge the gap between user descriptions and formal summaries. We integrated this model into a full-stack web application and evaluated it against a custom benchmark of 50 queries. Our results demonstrate that not only the keyword-only baselines but also our hybrid approaches satisfy users' demand effectively, successfully retrieving films even when user queries are vague or abstract.

1 Introduction

The number of movies available on streaming platforms today is overwhelming. While it is easy to find a movie if you know its title (e.g., Inception) or its star (e.g., Leonardo DiCaprio), it is surprisingly difficult to find a movie if you only remember the story. This report introduces UIUCinema, a project aimed at solving this common problem. We explore potential solutions to search with plot descriptions and propose a "hybrid" solution.

1.1 The Problem

Most movie search engines work like a phone book: they are excellent for looking up known entities but poor at handling descriptive queries. A user might vividly recall "that film where teenagers meet at a karaoke party and find out they go to the same school" but fail to remember the title High School Musical.

This failure originates from a fundamental disconnection between spoken language and database text. Users describe plots using natural, emotional, or conceptual language. Databases, however, use formal, structured summaries. A traditional search engine relies on exact word matches; if a user says "teenagers" but the summary says "adolescents," the search fails, even though the description is accurate.

1.2 Our solution: UIUCinema

Our goal was to build a system that bridges this "vocabulary gap" by understanding the meaning of a query, not just its keywords. To achieve this, we developed UIUCinema. Our contributions are:

- **Hybrid Search:** We recognized that neither keyword search nor "meaning-based" search is perfect in isolation. Keywords provide precision for names, while semantic search captures concepts. We combined them into a two-stage hybrid model: using BM25 to filter candidates and Sentence-Transformers to re-rank them based on narrative similarity.
- **Data Preprocessing:** We engineered an automated pipeline to enrich a dataset of 10,000 IMDb movie plots. By merging summaries with genre tags and keywords, we created detailed "super-documents" that maximize the system's ability to find matches.
- **Web Application:** We implemented our model within a responsive web application using Flask and TailwindCSS, demonstrating that complex semantic retrieval can be delivered in real-time.
- **Evaluation:** We evaluated our system against a custom test set of 50 diverse queries. Our experiments confirm that the hybrid model offers a robust solution to the plot retrieval problem, successfully handling queries where traditional methods fail.

2 Description

All of our code and completed work is available in our shared GitHub repository: https://github.com/TheoXiong7/UIUCinema. Our progress is shown in the following areas:

a) Data Preprocessing Pipeline

While the dataset provided a good foundation, the raw text fields contained unstructured natural language and varied formatting issues. To standardize the input, we built a robust preprocessing pipeline using Python's pandas and spaCy libraries (previous homeworks are quite helpful for us during this step).

- **Data Enrichment**: A single plot summary is often too short to capture everything. We merged the synopsis field with the genre, KeyBERT, YAKE, and SentenceTransformers keyword fields. This created a single, rich text field for every movie that contained both the story and important thematic tags.
- **Normalization**: We converted all text to lowercase and removed special characters, punctuation, and digits to standardize the input.
- **Lemmatization**: Instead of simple stemming, we used spaCy for lemmatization. This converts words to their root form (e.g., "studying" becomes "study", "running" becomes "run"), ensuring that different tenses of the same word are treated as matches.
- **Stopword Removal**: We filtered out common "stop words" (like "the", "and", "is") that add no meaning to the search. However, we were careful to keep negation words like "not" or "never", as these can change the entire meaning of a plot description.

b) Two-Stage Retrieval Pipeline

Our search engine employs a two-stage retrieve-then-rerank architecture that balances efficiency with effectiveness. This design allows us to leverage the speed of lexical search while benefiting from the semantic understanding of neural models.

Stage 1: BM25 Candidate Retrieval. The first stage uses the Okapi BM25 algorithm to quickly retrieve the top-k candidate documents from the entire corpus. BM25 is a probabilistic ranking function that scores documents based on term frequency, inverse document frequency, and document length normalization. We use the standard parameters $k_1 = 1.2$ and $b = 0.75$. This stage retrieves the top 200 candidates, dramatically reducing the search space for the more expensive semantic computation.

Stage 2: Semantic Reranking. The second stage reranks the BM25 candidates using dense semantic representations. We encode both the query and candidate documents using `all-MiniLM-L6-v2`, a SentenceTransformer model that maps text to 384-dimensional embeddings. The semantic similarity is computed via cosine similarity between the query and document embedding vectors.

Hybrid Score Fusion. The final ranking combines both signals using a weighted linear combination:

$$\text{score}_{\text{final}} = \alpha \cdot \text{sim}_{\text{semantic}} + (1 - \alpha) \cdot \text{score}_{\text{BM25}}$$

where α controls the balance between semantic and lexical matching. Both scores are min-max normalized before fusion to ensure they operate on the same scale.

Pseudo-Relevance Feedback (PRF). To address vocabulary mismatch between queries and documents, we optionally apply Pseudo-Relevance Feedback. PRF assumes that the top-k documents from an initial retrieval are relevant and extracts high-IDF terms from them to expand the original query. This helps when users describe plots using different words than those in the synopsis.

c) Hyperparameter Tuning

We used Optuna, a Bayesian optimization framework, to tune four key hyperparameters over 30 trials, maximizing nDCG@10 on our evaluation query set.

Parameter Definitions.

- α (Fusion Weight): Controls the balance between semantic and lexical scores in the hybrid ranking formula. Higher values favor neural similarity; lower values favor BM25.
- w_{plot} (Synopsis Weight): Determines the importance of the plot synopsis field in BM25F scoring, affecting how much exact text matches in the summary contribute to retrieval.
- w_{kw} (Keyword Weight): Assigns importance to the extracted keyword fields (from KeyBERT, YAKE, and SentenceTransformers), boosting documents that share thematic terms with the query.
- w_{meta} (Metadata Weight): Governs the contribution of genre information to the BM25F relevance calculation.

The optimal values are shown in Table 1. Notably, the high w_{meta} value (0.54) suggests that genre matching is particularly discriminative for movie retrieval, while the α value of 0.70 indicates a preference for semantic understanding over pure lexical matching—consistent with our use case of natural language plot descriptions.

Table 1: Optimal Hyperparameter Values

Parameter	Value
α	0.6950
w_{plot}	0.3265
w_{kw}	0.1297
w_{meta}	0.5438

d) Web Application

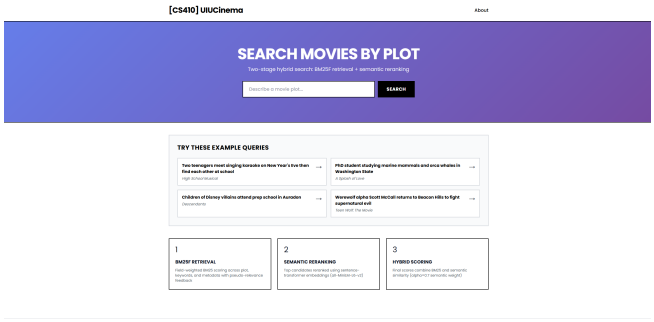

Figure 1: Landing Page

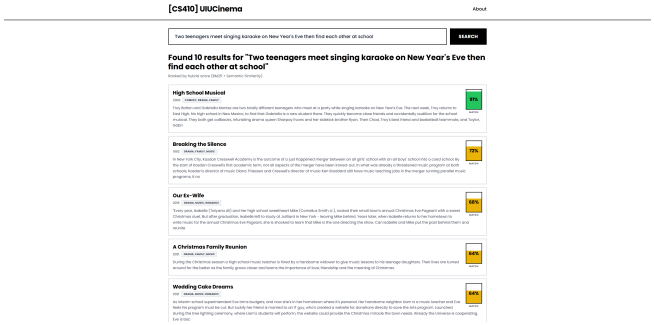

Figure 2: Search Results Page

We designed UIUCinema as a full-stack web application to demonstrate our search model in a real-world setting (Figure 1 and 2).

- **Backend (Flask):** We chose Flask (Python) for our server because it integrates seamlessly with our data science libraries. When the server starts, it loads our pre-computed search index and the heavy semantic model into memory. This ensures that when a user searches, the results are returned instantly without reloading the model.
- **Frontend (TailwindCSS):** The user interface was built using HTML and TailwindCSS. The results page displays the movie title, a snippet of the plot, and a relevance score, giving users insight into why a movie was selected.

3 Evaluation

To assess the performance of UIUCinema, we conducted a two-part evaluation: a statistical analysis of our dataset quality and a comparative experiment of our retrieval models against a custom benchmark of 50 queries.

3.1 Data Analysis

Before evaluating the models, we analyzed the quality of our preprocessed corpus (9,746 movies) to ensure it was suitable for retrieval tasks.

3.1.1 Text Length Statistics. Our analysis of the preprocessed plot summaries revealed:

- **Average Plot Length:** 75.11 words (436.96 characters)
- **Median Plot Length:** 70 words (400 characters)
- **Longest Plot:** 1640 words
- **Shortest Plot:** 2 words
- **Average Processed Text:** 66 words after cleaning and lemmatization

The average length of **75.11** words suggests the summaries are sufficiently detailed for retrieval, though the significant reduction to 66 words after preprocessing (12% reduction) indicates substantial removal of stopwords and non-content terms. The relatively tight distribution around the median suggests consistent synopsis quality across the dataset.

3.1.2 Temporal and Genre Distribution. The dataset spans from 1937 to 2024, providing nearly a century of film data. However, the distribution is heavily skewed toward recent decades. Genre analysis reveals significant class imbalance:

- **Drama:** 3,158 movies (32.4%)
- **Comedy:** 1,685 movies (17.3%)
- **Action:** 759 movies (7.8%)
- **Crime:** 743 movies (7.6%)
- **Documentary:** 712 movies (7.3%)

This genre imbalance means that drama and comedy films are over-represented in our corpus, which could potentially bias search results toward these genres. The top 5 genres account for approximately 72% of the dataset.

3.1.3 Vocabulary Analysis. After our cleaning pipeline (lemmatization and stopword removal), we analyzed the term frequencies to verify data quality. The most common terms include content-rich words such as "love" (10,841 occurrences), "family" (8,087), "life" (7,796), and "man" (6,831), confirming that the preprocessing successfully retained meaningful vocabulary while removing noise. Domain-specific terms like "Christmas", "murder", and "father" also appear frequently, indicating the corpus captures diverse movie themes.

3.2 Test Queries

We created 50 queries to test our model's performance given queries of different quality.

Below is an example subsection of the queries used in our evaluation, along with the corresponding relevant movie(s). Each query is associated with a relevance score indicating the ground truth for evaluation metrics.

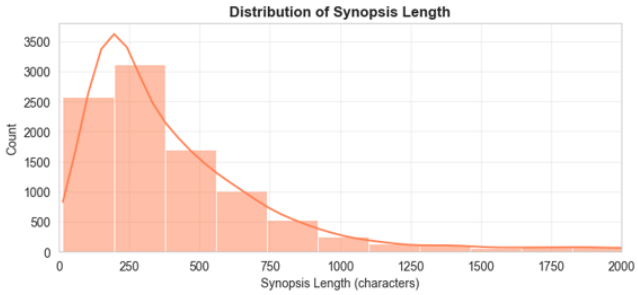

Figure 3: Synopsis Length Distribution

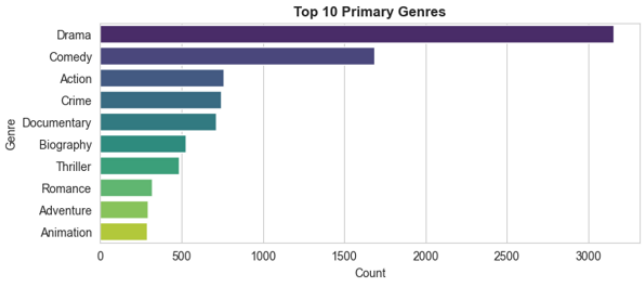

Figure 4: Top Genres

- {"query": "cheerleader falls for a zombie boy in a school divided by humans and zombies", "relevant": {"Z-O-M-B-I-E-S": 2}}
- {"query": "adult Scott McCall reunites with his pack to fight a new supernatural evil", "relevant": {"Teen Wolf: The Movie": 2}}
- {"query": "kids of famous Disney villains go to a prep school with heroes' children", "relevant": {"Descendants": 2}}
- {"query": "movie where a marine biology student falls for a whale-watching tour guide", "relevant": {"A Splash of Love": 2}}

For each query, the model retrieves a ranked list of movies.

3.3 Model Evaluation

We tested our model on the 50 test queries and, using the retrieval results, calculated the following metrics:

- Precision@10 (P@10): Measures the proportion of relevant movies among the top 10 retrieved results, reflecting immediate relevance.
- nDCG@10: Normalized Discounted Cumulative Gain at 10, which assesses the ranking quality of the top 10 results, giving higher weight to relevant movies appearing earlier.
- Success@3: Indicates whether at least one relevant movie appears within the top 3 results, capturing the likelihood that a user finds a relevant movie quickly.
- MRR (Mean Reciprocal Rank): Evaluates the ranking position of the first relevant result, with higher values indicating that relevant movies are retrieved near the top of the list.

We compared the performance of three retrieval models: BM25 only, BM25 combined with semantic embeddings (hybrid search),

Table 2: Comparison of Retrieval Models

Model	P@10	nDCG@10	Success@3	MRR
BM25 Only	0.049	0.365	0.367	0.325
BM25 + Semantic	0.055	0.399	0.408	0.352
BM25 + Semantic + PRF	0.055	0.368	0.367	0.312

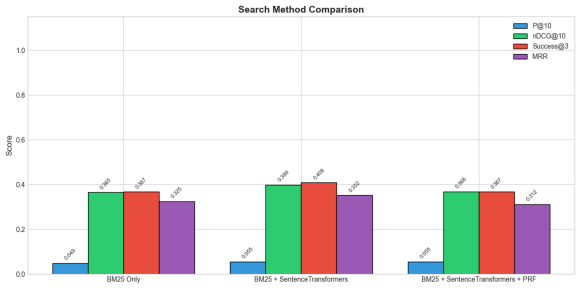

Figure 5: Model Comparison: BM25 vs Hybrid Search vs Hybrid Search + PRF. Metrics shown include Precision@10, nDCG@10, Success@3, and MRR.

and hybrid search enhanced with pseudo-relevance feedback (PRF). See Figure 5. Across all metrics, the hybrid model (BM25 + Semantic) outperformed BM25 alone, demonstrating that incorporating semantic similarity improves both relevance and ranking quality. This indicates that the hybrid approach is more effective at capturing relevant movies even when user queries do not match keywords exactly.

Interestingly, adding PRF to the hybrid model did not yield further improvements and, in some metrics, slightly reduced performance. This suggests that the simple PRF strategy may have introduced noise or irrelevant expansions, offsetting the benefits of semantic matching. Overall, the hybrid model without PRF offers the best balance between relevance and ranking quality for descriptive movie queries.

4 Discussion

4.1 Example Search Results

The tables below show example results from the hybrid search model with tuned parameters on our test queries:

Table 3: Search Results for Disney/Musical Queries

Query	In Top 3 Result	Score
teenagers audition for a school musical while balancing sports and friendships	High School Musical	1.000
kids of famous Disney villains go to a prep school with heroes' children	Descendants 3	0.976
cheerleader falls for a zombie boy in a school divided by humans and zombies	Z-O-M-B-I-E-S	0.978

The first table demonstrates that our model successfully retrieves the correct movie as the top result for straightforward queries. The second table shows a case where a German film with a similar zombie-cheerleader theme slightly outranks the expected result,

Table 4: Top-5 Results for Query: "cheerleader falls for a zombie boy..."

Rank	Movie Title	Score
1	Was soll bloß aus dir werden	0.994
2	Z-O-M-B-I-E-S	0.978
3	The Marriage Fool	0.762
4	Ice Sculpture Christmas	0.758
5	Nuclear Family	0.745

illustrating both the model's semantic understanding and the challenge of handling multiple relevant movies in the corpus.

4.2 Future Potential Improvements

We've identified realistic limitations that we will need to address:

- **Dataset Limitations:** Although our preprocessed corpus contains 9,746 movies—already a large number—it does not cover the entire cinematic universe. We observed that a moderate number of films are absent from the dataset, and some could be popular. Consequently, we recommend the staff input plot descriptions for movies known to be within the dataset to ensure valid retrieval results. In future development, we plan to address this coverage gap by integrating larger, more comprehensive datasets, such as the CMU Movie Summary Corpus, to encompass a wider range of films. While scaling to such datasets introduces significant computational complexity, addressing these performance challenges is a critical and exciting next step for the project's improvement.
- **Model Limitations:** Despite the improvements offered by the hybrid model, the system remains sensitive to query phrasing. We observed that some highly abstract, metaphorical, or culturally nuanced queries can still fail to retrieve the correct movie. Additionally, queries that describe multiple disjoint plot elements or complex character interactions in unusual ways can occasionally confuse the re-ranking mechanism, reducing accuracy.

References

[1] Venkata Praneeth Villuri. 2023. *IMDB Top 10000 movie plots' keywords.* Kaggle. https://www.kaggle.com/dsv/5128589.