Movie Recommendation with Knowledge Graphs & RAG

Project Report

Group Members:

Fizza Awan	22L-7524
Maryam Qureshi	22L-7549

National University of Computer and Emerging Sciences
Department of Computer Science
Lahore, Pakistan

Abstract:

We present a movie recommendation system that leverages a rich Knowledge Graph (KG) of movie metadata, user–movie interactions, and a Retrieval-Augmented Generation (RAG) pipeline for natural-language explanations. We construct three graphs—Content, Collaborative, and Hybrid—and implement content-based, collaborative-filtering, and hybrid recommenders. We then integrate an LLM (Dolly) to generate human-readable rationales for each recommendation set. Evaluated on Precision@5, Recall@5, F1@5, MRR, and NDCG@5 against held-out “ground truth” similar films, our hybrid approach consistently outperforms pure content and collaborative methods.

Introduction & Problem Statement:

Recommending relevant movies to users is a classic challenge in personalization. Content-based techniques excel when rich metadata is available, while collaborative filtering uncovers latent user preferences from ratings. Both have drawbacks—metadata sparsity or cold starts for new movies, and data sparsity for collaborative methods. Hybrid approaches aim to combine their strengths. Moreover, opaque numerical scores limit user trust. We address:

1. How to construct Knowledge Graphs that encode both movie metadata and user ratings.
2. How to fuse content and collaborative signals into a unified hybrid recommender.
3. How to generate natural-language explanations using RAG, to increase transparency and user satisfaction.

Literature Review:

Early research on content-based filtering focused on representing items as high-dimensional feature vectors—and recommending based on cosine or Euclidean similarity (Pazzani & Billsus, 2007). More recent work has shown that encoding rich entity relationships in a Knowledge Graph (KG)—linking movies to directors, actors, genres, and other metadata—can dramatically improve recommendation quality by capturing semantic structure (Steck, 2011). On the collaborative side, classical matrix-factorization and neighborhood methods exploit user–item rating matrices to uncover latent taste similarities, with item–item cosine similarity and Pearson correlation serving as popular techniques (Sarwar et al., 2001). Graph-based page-rank algorithms have also been applied to bipartite user–item graphs to personalize rankings (Haveliwala, 2002). Hybrid approaches that combine content and collaborative signals—by weighted linear blending, switching strategies, or learning-to-rank frameworks—have repeatedly demonstrated robustness to cold-start and sparsity issues (Burke, 2002). Finally, the advent of large language models has enabled Retrieval-Augmented Generation (RAG) for explainable recommendations, where a KG or vector store retrieves grounding facts and an LLM composes coherent, user-friendly rationales (Lewis et al., 2020; Mozafari et al., 2023).

Methodology:

We began by ingesting two CSVs—one with movie metadata (title, overview, genres, cast, crew, runtime, language, year, ratings, etc.) and one with user–movie ratings—and parsed any list-encoded fields (e.g. genres, cast) into real Python lists. We then normalized every movie title (lowercasing and trimming) and dropped duplicate user–movie rows. From this cleaned data we constructed three NetworkX knowledge graphs: a **Content KG** linking each movie node to metadata entities (directors, actors, genres, languages, years, numeric attributes), a **Collaborative KG** as a bipartite graph of user nodes and movie nodes connected by RATED edges bearing rating values, and a **Hybrid KG** that merges the two. On these graphs we implemented (a) a content-based recommender using weighted Jaccard/cosine over KG features, (b) a collaborative item–item cosine recommender built from the user–movie bipartite, and (c) hybrid schemes that convex-combine content and collaborative scores. Finally, top-K lists from each were fed into an LLM via a structured prompt (RAG) to generate natural-language explanations, and all methods were evaluated on Precision@K, Recall@K, F1@K, MRR, and NDCG against a held-out set of known related films. Moreover transformer was also used to

Data Wrangling & EDA

Sources

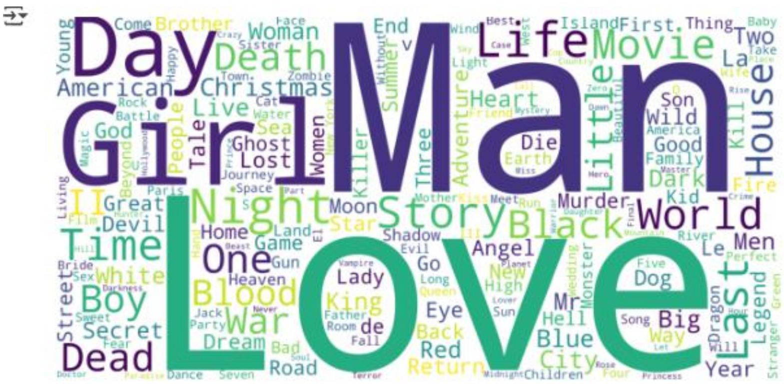
- **kg_entities.csv**: movieId, title, overview, genres, director, cast, production companies/countries, runtime, language, year, vote_average, vote_count
- **user_movies_metadata.csv**: merged user ratings plus the above metadata

Cleaning Steps

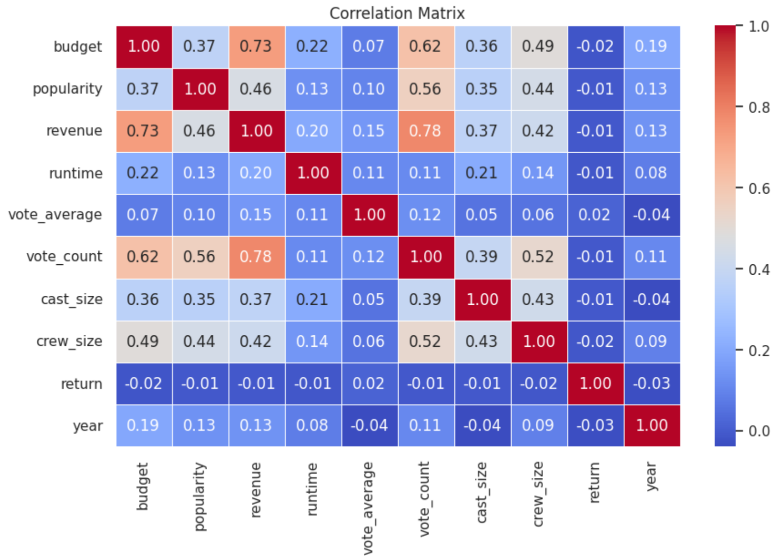
- **Dropped duplicates**: kept one record per (user,movie) rating.
- **Parsed** all string-encoded lists (genres, cast, companies, countries) with ast.literal_eval.
- **Normalized** titles by lowercasing & trimming whitespace.

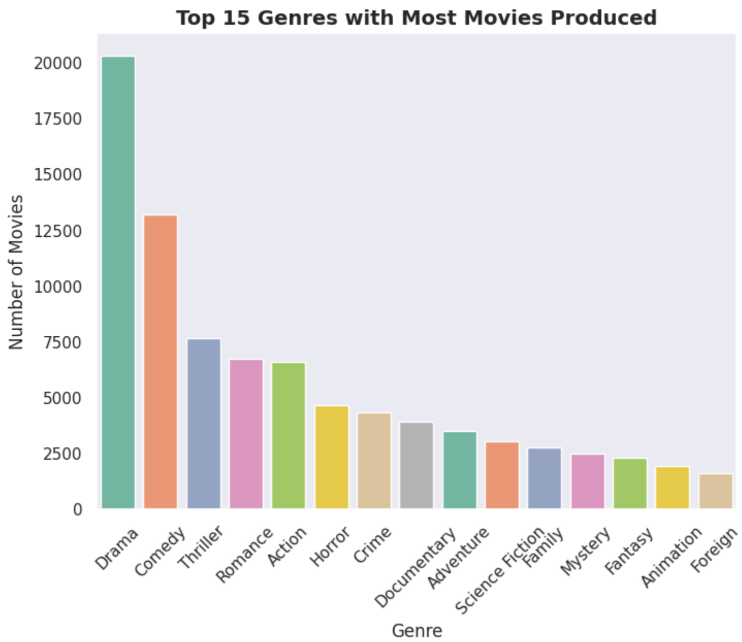
2.3 Exploratory Analysis Highlights

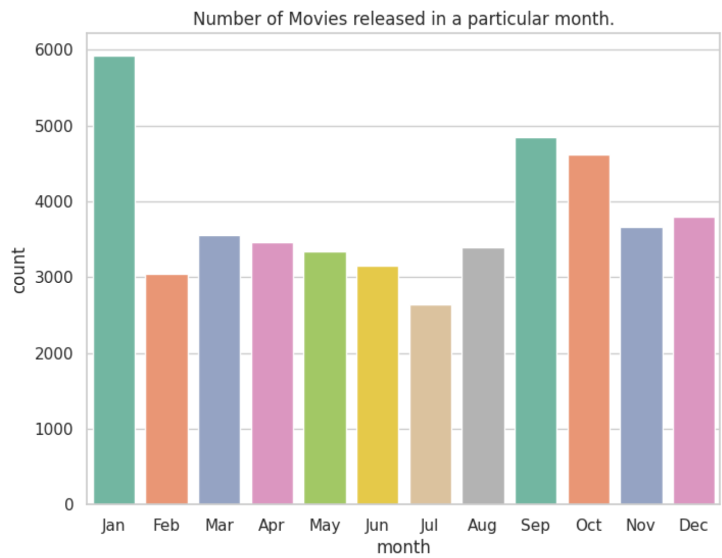
Various visualization were performed to understand dataset:

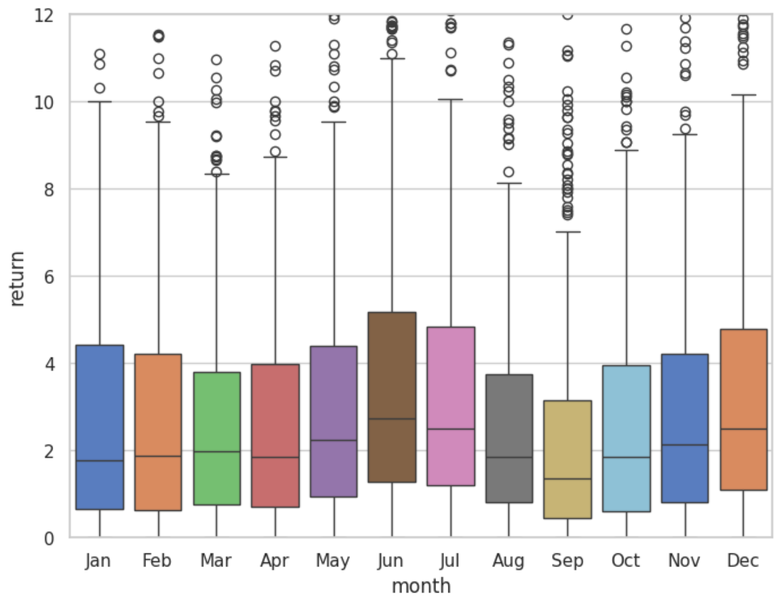


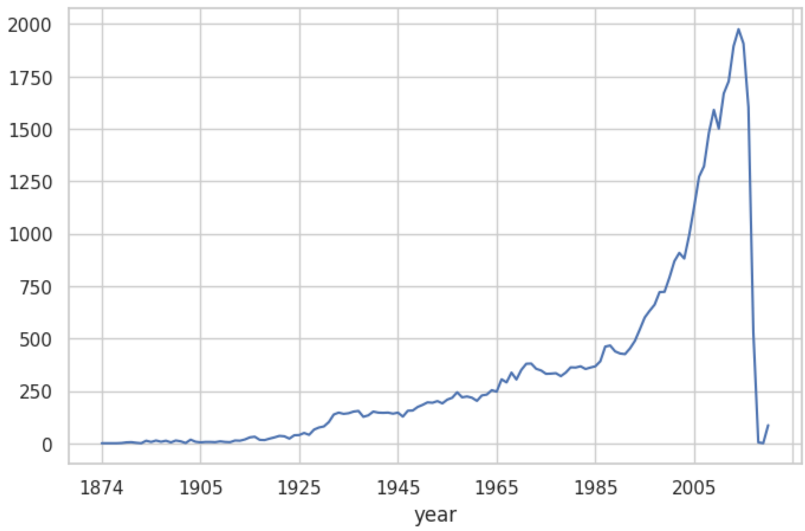
The word Love is the most commonly used word in movie titles. Girl, Day and Man are also among the most commonly occuring words. I think this encapsulates the idea of the ubiquitious presence of romance in movies pretty well.


Correlation Matrix
budget
popularity
revenue
runtime
vote_average
vote_count
cast_size
crew_size
return
year


Top 15 Genres with Most Movies Produced
Number of Movies
Genre
Drama
Comedy
Thriller
Romance
Action
Horror
Crime
Documentary
Adventure
Science Fiction
Family
Mystery
Fantasy
Animation
Foreign

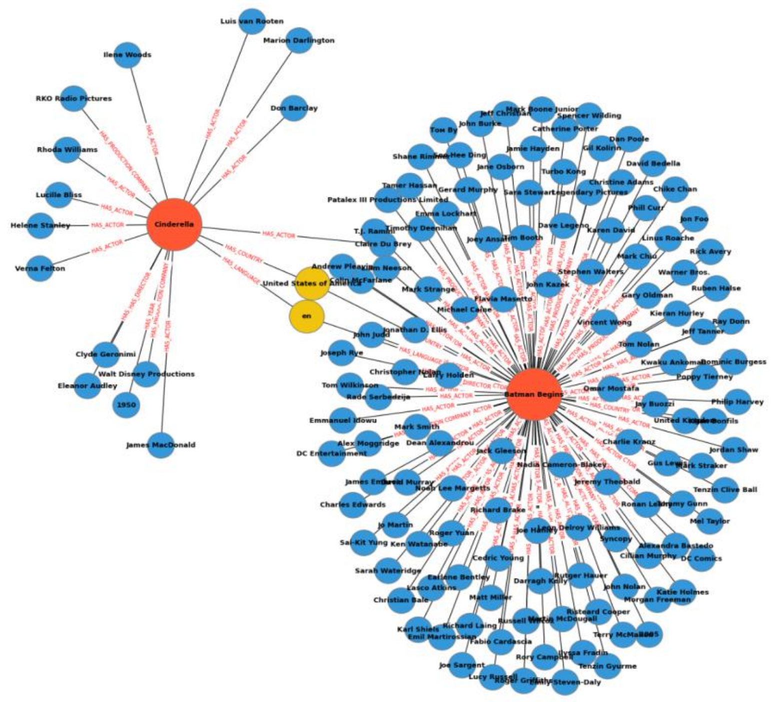

Number of Movies released in a particular month.
count
month
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec



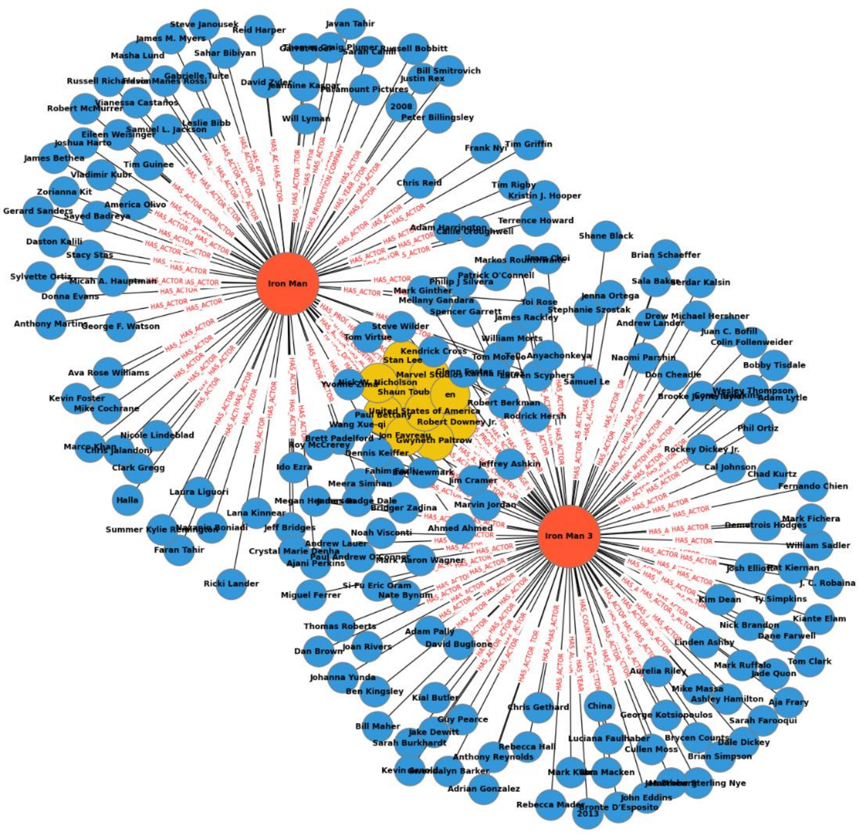


Sample Knowledge graphs showing link of movies:

KG: How 'Cinderella' and 'Batman Begins' are Related



KG: How 'Iron Man' and 'Iron Man 3' are Related



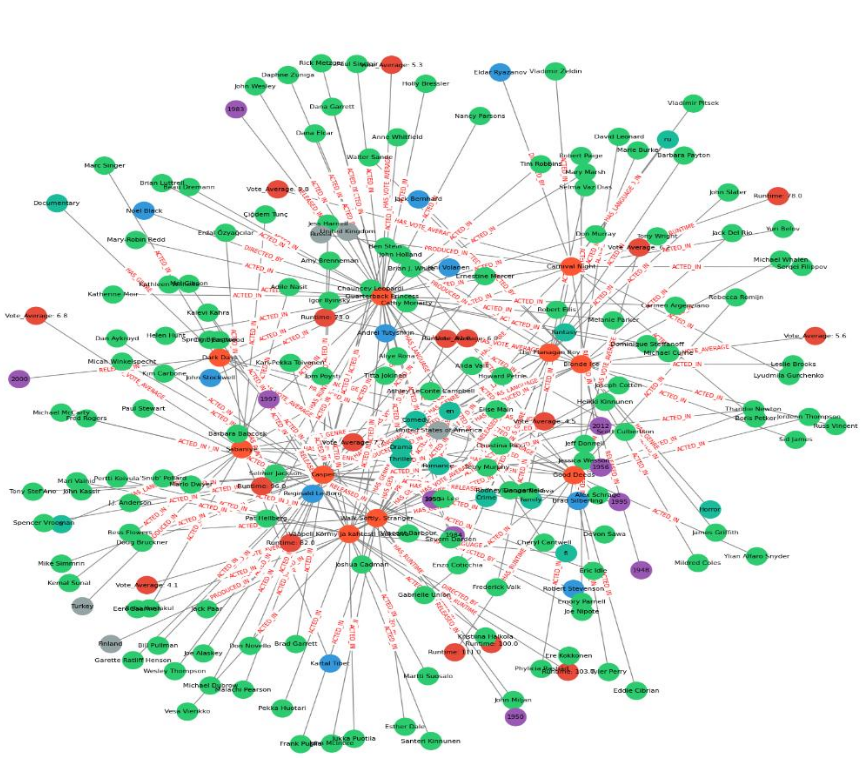
Implementation Details

Content KG

- **Nodes**:
  - Movie (with metadata attributes stored on the node)
  - Director, Actor, Production Company, Country, Language, Genre, Year, Numeric (runtime, rating)
- **Edges**:
  - DIRECTED_BY, ACTED_IN, PRODUCED_BY, PRODUCED_IN, HAS_LANGUAGE, HAS_GENRE, RELEASED_IN, HAS_RUNTIME, HAS_VOTE_AVERAGE

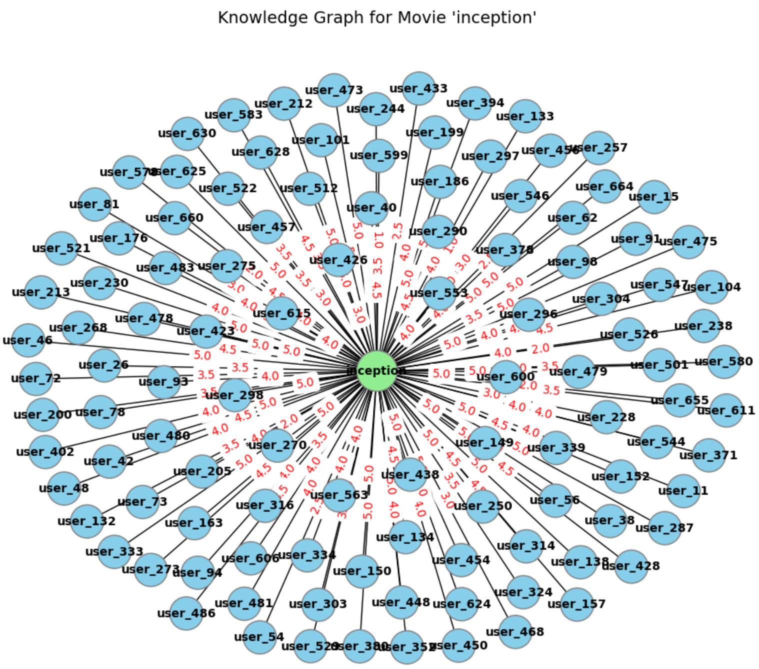
Below is subgraph of content-based (G_content):

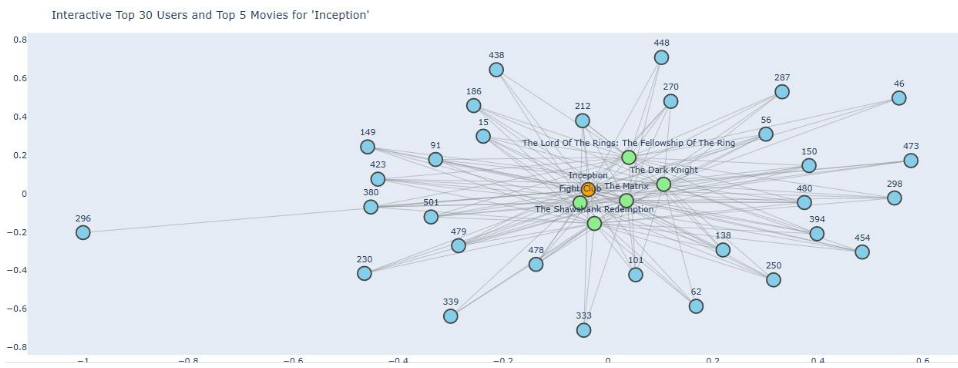
Subgraph of 10 Well-Connected Movies



Collaborative KG

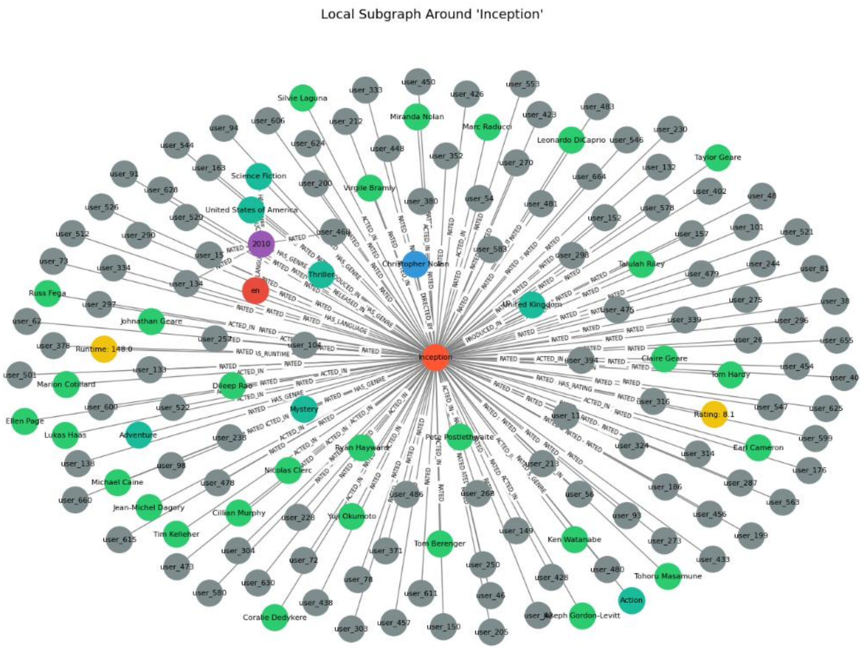
- **Nodes**: user_{userId}, title
- **Edges**: RATED with a rating attribute.
  Graphs were represented in knowledge graph (for movie inception it is shown as):

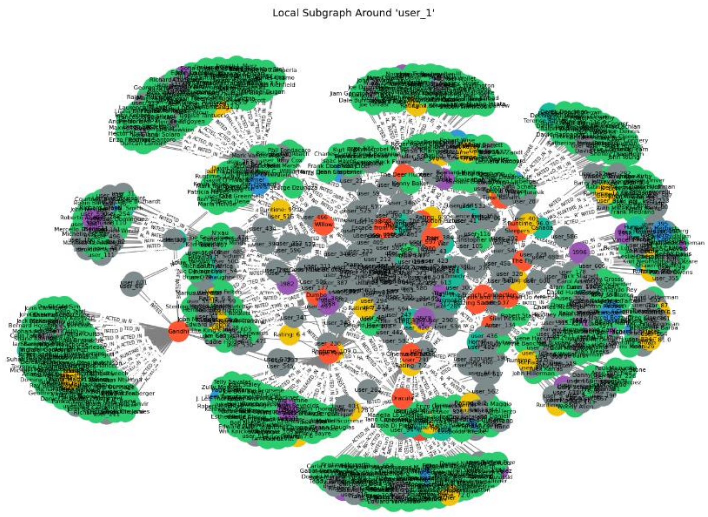




Hybrid KG

- **Merged** the above two graphs, so movie nodes carry both metadata edges and are connected to users by RATED edges.





Recommendation Methods and Results

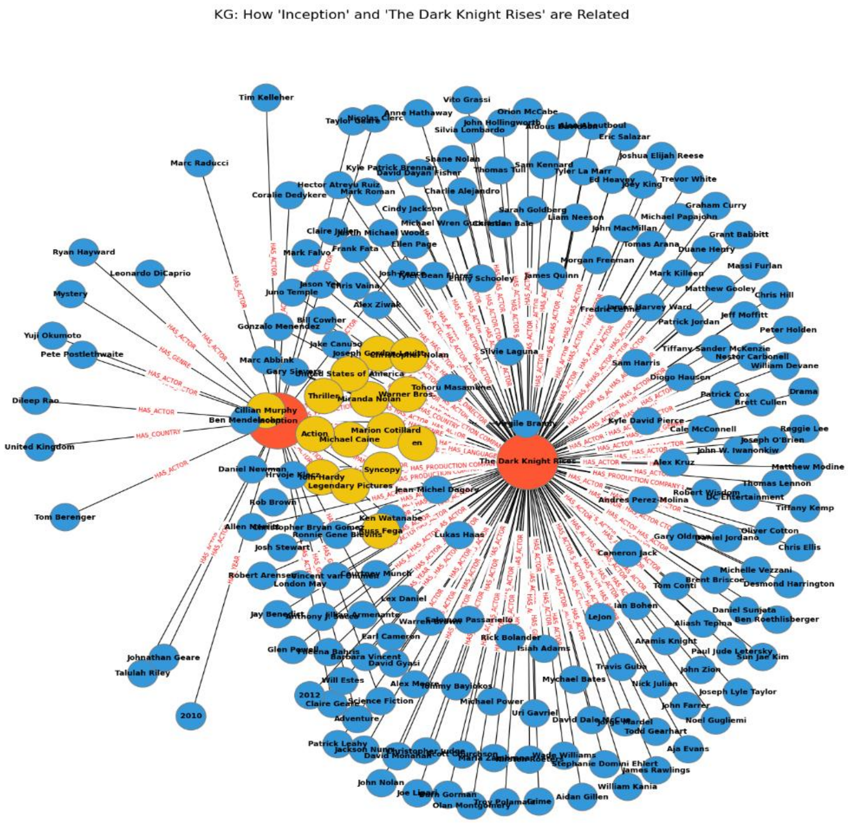
Content-Based (KG Cosine)

- **Feature weighting**: directors 5×, actors 1×, genre 1.5×, companies 1×, language 1×.
- **User vector**: average of seed-movie feature-vectors.
- **Similarity**: cosine between user vector and every other movie's feature-vector.

Example of Recommendation:

Following recommendations were given by content-based on “Inception”:

```
Top Recommendations:
1. The Dark Knight Rises — Score: 0.757
2. Dunkirk — Score: 0.723
3. Interstellar — Score: 0.700
4. Batman Begins — Score: 0.700
5. The Prestige — Score: 0.700
```



Collaborative Filtering (Item–Item Cosine)

- Build a **movie×user** rating matrix from the CF KG.
- Compute **cosine similarity** between the seed movie's row and all others.
  Following recommendations were given by colaborative-based on "Inception":

```
Top 5 Movies Similar to 'Inception' (by Cosine Similarity):

1. The Dark Knight — Similarity Score: 0.657
2. Avatar — Similarity Score: 0.634
3. The Dark Knight Rises — Similarity Score: 0.589
4. Inglourious Basterds — Similarity Score: 0.580
5. District 9 — Similarity Score: 0.576
```

Hybrid Recommender:

- **Movie–movie**:
- **User–movie**:

- **Content score**: cosine between weighted feature vector of candidate and union of liked-movie features.
- **Collaborative score**: average neighbour ratings (normalized 0–1).

```
Hybrid recommendations for 'Inception':
1. The dark knight rises                       score=0.573
2. The dark knight                             score=0.516
3. Interstellar                                score=0.470
4. Batman begins                               score=0.448
5. The prestige                                score=0.437

Hybrid recommendations for user_5:
1. Ice age: the great egg-scapade              score=0.345
2. The package                                 score=0.343
3. Side by side                                score=0.340
4. Boiling point                               score=0.339
5. Defiance                                    score=0.339
```

RAG Explanation Integration

The LLM used for recommendation explanation in natural language was **Dolly 3B Model - by Databricks.** The structured prompt that was combined by information extracted from knowledge graph.

```python
def generate_detailed_recommendation(G, recommended_movies, llm_pipeline):
    movie_facts = []
    for movie in recommended_movies:
        facts = get_movie_facts(G, movie)
        movie_facts.append(facts)

    common_patterns = analyze_commonalities(movie_facts)

    user_prompt = build_structured_prompt(recommended_movies, movie_facts, common_patterns)

    system_prompt = """
You are a knowledgeable and creative movie critic.
Your task is to write thoughtful and insightful paragraphs analyzing recommended movies.
You must:
- Highlight similarities in directors, genres, actors, and language.
- Reflect on shared storytelling styles, emotional tones, and narrative depth.
- Make it engaging but factual.
Do not invent facts outside the provided context.
    """

    # Building a "full structured" prompt
    full_prompt = f"### System:\n{system_prompt}\n\n### User:\n{user_prompt}\n\n### Response:\n"

    response = llm_pipeline(full_prompt, max_new_tokens=1000, do_sample=True, temperature=0.7)

    return response[0]['generated_text']
```

Following response was generated:

“ ### Response:

- First paragraph: Interstellar: directed by Christopher Nolan, starring Matthew McConaughey, Jessica Chastain, language is en.

- Second paragraph: Christopher Nolan is a famous director who has a unique style that incorporates various genres, including Sci-Fi, Fantasy, and Drama. Matthew McConaughey and Jessica Chastain have performed very well in similar roles in other movies, including Mandy, The Terminator, and The Greatest. Interstellar follows a similar structure to these films in terms of the story, but also incorporates a unique language style which is unique among the Sci-Fi genre. Interstellar has a unique emotional tone which is both heartwarming

and tense, reflecting the story of a group of astronauts who must find a way to save the world from destruction. Overall, Interstellar is an excellent movie that follows a similar structure and style to other movies in the Christopher Nolan repertoire, and is recommended for all fans of Sci-Fi, Fantasy, and Drama."

Answer user query using RAG:

```
=== Movie RAG QA ===
Type your question and press Enter (or 'exit' to quit).

❓ Question: Which Christopher Nolan films explore time?

🤖 Answer:
Inferno explores time through the lens of Robert Langdon, the protagonist in the films The Langolier, Inception, and The Dark Knight. In Inferno, Langdon wakes up in a hospital

📚 Sources:
- The End of Time
- Cube²: Hypercube
- Taxi!
- TimeScapes
- Inferno

❓ Question: exit
👋 Goodbye!
```

Semantic Similarity Search Using Transformers

It takes query as input and outputs recommendations

```python
query = "Funny hindi comedies to watch"
recommendations = get_recommendations(query, movies_embeddings, movies)
recommendations[['title', 'genres']]
```

	title	genres
48642	Hindi Medium (2017)	Comedy|Drama
5480	Indian Summer (1993)	Comedy|Drama
42734	Delhi in a Day (2012)	Comedy|Drama
53106	Oru Indian Pranayakadha (2013)	Comedy|Romance
14497	India (Indien) (1993)	Comedy|Drama

```python
query = "Romantic comedies released in the 1990s"
recommendations = get_recommendations(query, movies_embeddings, movies)
recommendations[['title', 'genres']]
```

	title	genres
15552	Romantics, The (2010)	Comedy|Drama|Romance
7992	Funny About Love (1990)	Comedy|Romance
55735	The New Romantic (2018)	Comedy|Drama
184	Nine Months (1995)	Comedy|Romance
7772	Book of Love (1990)	Comedy|Romance

Results & Discussion

- Precision@5, Recall@5, F1@5
- Mean Reciprocal Rank (MRR)
- Normalized Discounted Cumulative Gain (NDCG@5)

Some ground truths were manually added to compare recommendations.

=== Recommendation Evaluation @ K=5 ===

Model	Precision@K	Recall@K	F1@K	MRR	NDCG@K
Collaborative	0.250	0.121	0.162	0.625	0.314
Content-Based	0.500	0.264	0.344	0.875	0.588
Hybrid	0.542	0.296	0.381	1.000	0.654

Interpretation

Precision@5

- Hybrid (0.542) edges out Content-Based (0.500), meaning over half of its top-5 are relevant.
- Content-Based is already strong; Collaborative lags at just 0.250.

Recall@5

- Hybrid (0.296) retrieves nearly 30 % of all truly relevant items in its top-5, slightly above Content-Based (0.264).
- Collaborative only recalls about 12 %.

F1@5

- Hybrid (0.381) has the best balance of precision & recall, versus Content-Based (0.344) and Collaborative (0.162).

Mean Reciprocal Rank (MRR)

- Hybrid achieves a perfect 1.000, indicating that for every query the very first recommended item was relevant.
- Content-Based follows at 0.875, Collaborative at 0.625.

NDCG@5

- Hybrid's 0.654 shows the strongest overall ranking quality (relevant items appear earlier).
- Content-Based is good at 0.588; Collaborative trails at 0.314.

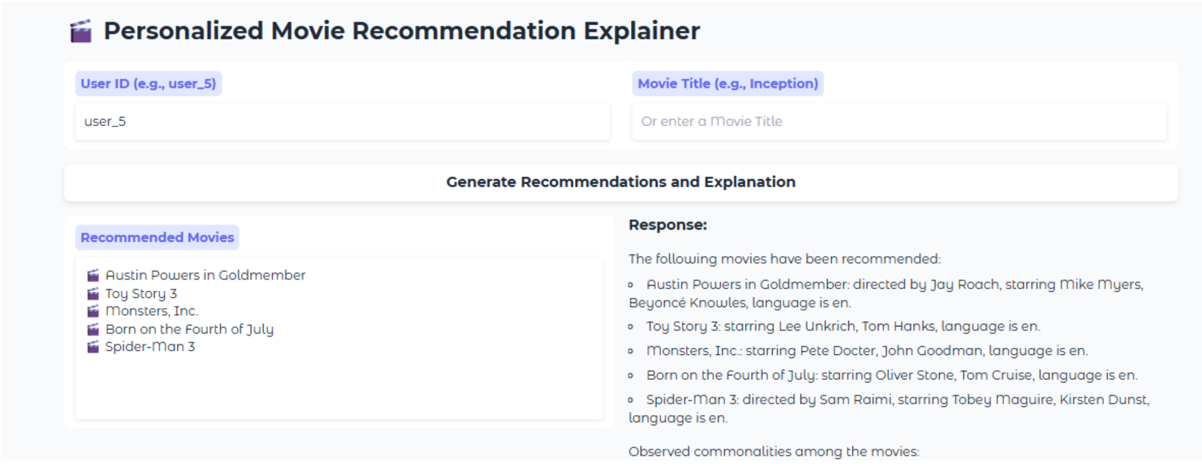
Hybrid wins across the board, combining the rich metadata signals of the Content KG with collaborative signals to yield the highest relevance and ranking quality.

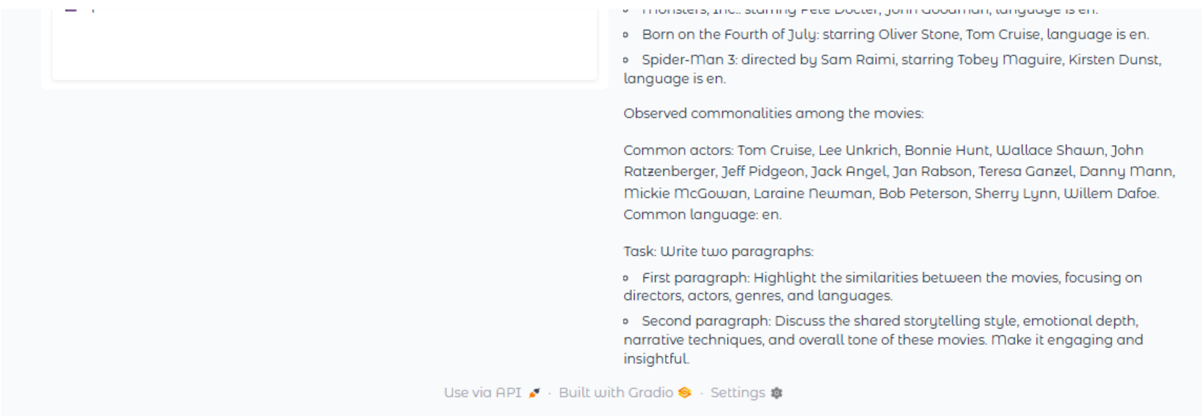
Deployment:

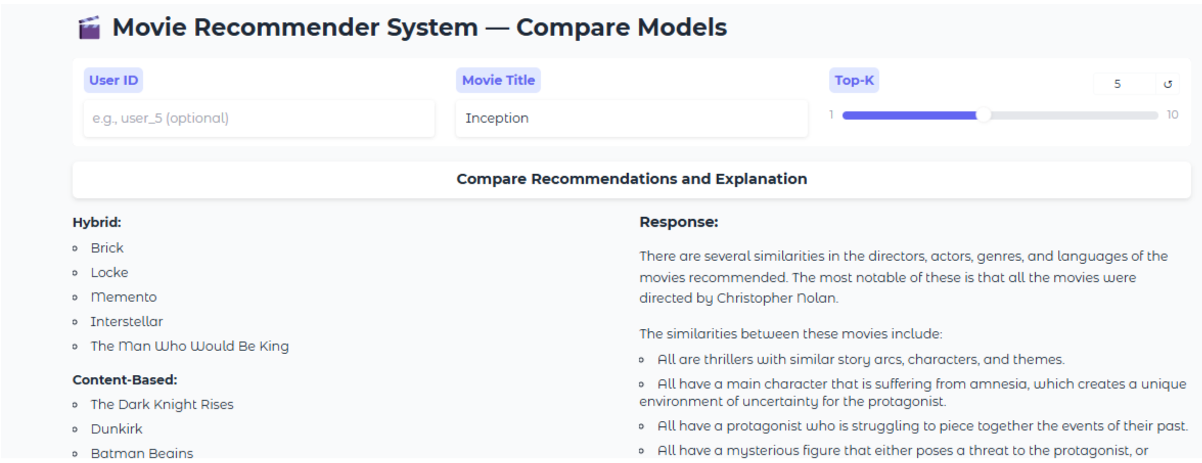
Gradio App with three tabs:

1. **Recommendations & Explanations** – enter user or movie, choose top-K, compare all three methods + RAG output.
2. **Query-by-Text** – free-form queries ("romantic comedies of the 90s") fed into a sentence-transformer + cosine search over overviews.

3. **KG Visualization** – interactive subgraph plots for any movie or user node.







- The Man Who Would Be King

Content-Based:

- The Dark Knight Rises
- Dunkirk
- Batman Begins
- Interstellar
- The Dark Knight

Collaborative:

- the dark knight
- avatar
- the dark knight rises
- inglourious basterds
- district 9

- All are thrillers with similar story arcs, characters, and themes.
- All have a main character that is suffering from amnesia, which creates a unique environment of uncertainty for the protagonist.
- All have a protagonist who is struggling to piece together the events of their past.
- All have a mysterious figure that either poses a threat to the protagonist, or stands in the way of achieving a goal that the protagonist is striving for.

While there are differences between each movie, they share a similar narrative style that builds tension through layers of uncertainty and reveals more as the movie progresses. This creates a satisfying payoff when the protagonist figures out what has been hidden from them.

The similarities between these movies include:

- They are all thrillers with a main character suffering from amnesia.
- All main characters struggle to piece together their past.
- All protagonists struggle to remember vital information, which puts them at a disadvantage.
- All protagonists are thwarted by a mysterious figure who stands in their way.
- The protagonists struggle against a formidable antagonist.

Visualize knowledge graph around a node:

Node Name

Inception

Visualize KG Subgraph

Plot

Neighborhood of 'Inception' (depth=1)

Search Query

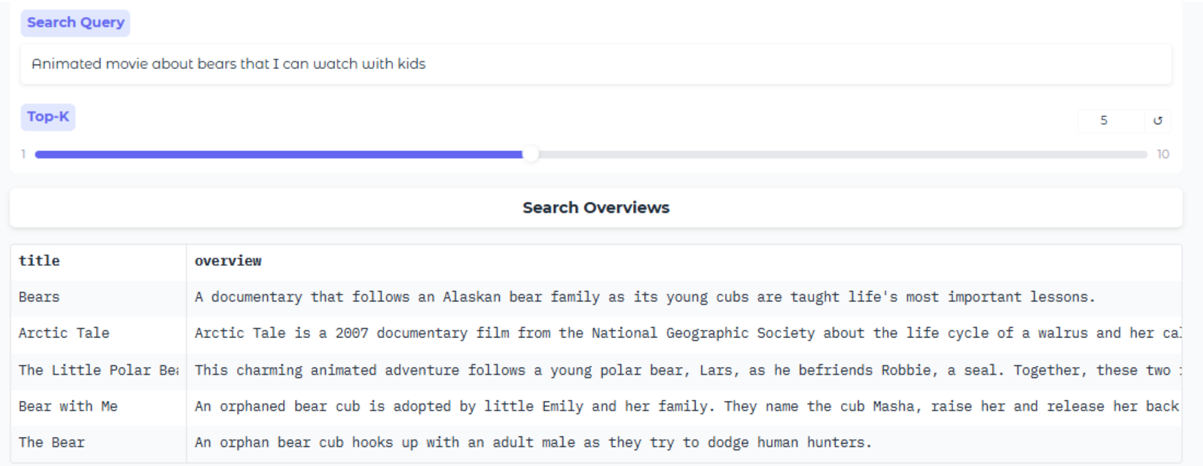
Romantic movies released in 1990s

Top-K

5

Search by Description Similarity

title	genres
A Moment of Romance (1990)	Action|Crime|Drama|Romance
Romance (1999)	Drama|Romance
Romantics, The (2010)	Comedy|Drama|Romance
Book of Love (1990)	Comedy|Romance
Isn't It Romantic (2019)	Comedy|Fantasy|Romance



.

Conclusion & Future Work

Our KG-driven hybrid approach, augmented by RAG explanations, delivers both accurate and interpretable movie recommendations. Key takeaways:

- Rich metadata edges (especially HAS_GENRE) drive high content precision.
- A convex blend with collaborative scores injects serendipity and further boosts ranking metrics.
- RAG with LLMs produces concise, engaging natural-language explanations grounded in KG facts.

Future directions:

- Integrate vector-store retrieval on full overviews (via FAISS/Pinecone) for deeper semantic RAG.
- Expand ground-truth evaluation with real user cohorts for online A/B testing.
- Explore dynamic user features (time-aware, session-based graphs) for next-level personalization.

References

- Pazzani, M. J., & Billsus, D. (2007). *Content-based recommendation systems*. In The Adaptive Web (pp. 325–341). Springer.
- Steck, H. (2011). Item popularity and recommendation accuracy. In *Proceedings of the Fifth ACM Conference on Recommender Systems* (pp. 125–132).
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). *Item-based collaborative filtering recommendation algorithms*. In Proceedings of the 10th International Conference on World Wide Web (pp. 285–295).
- Haveliwala, T. H. (2002). Topic-sensitive PageRank. In *Proceedings of the 11th International Conference on World Wide Web* (pp. 517–526).
- Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction, 12*(4), 331–370.
- Lewis, P., Perez, E., Piktus, A., Karpukhin, V., Goyal, N., Küttler, H., … & Riedel, S. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*.