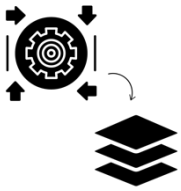
The British University in Egypt



NLP
Multi-Agent Movie Recommender System

Faculty of Informatics and Computer Science

Semester: Two

Academic Year: 2024/2025

Name	ID
Mostapha Abdelaziz Abdullah	227824
Abdelrahman Gamal	227998
Mahmoud Hussein	221523
Ahmed Imad	226061

Declaration of Academic Honesty

We declare that this report is my own work and that all sources used are properly acknowledged.

Signature: Ahmed Imad, Mahmoud Hussien, Abdelrahman Gamal, Mostapha Abdulaziz

Date: May 03, 2025

Table of Contents

1. Introduction

The implementation of this project developed a multi-agent movie recommendation system which generates specific movie recommendations by analyzing user requests in their specific contexts. The system manages its subgraphs through LangGraph in a hierarchical manner to execute agents

who perform recommendation production and personalized contextual recommendations while creating explanations for user suggestions. The recommendation system utilizes the google/flan-t5-small LLM obtained through Hugging Face pipeline while generating text embeddings with sentence-transformers/all-MiniLM-L6-v2. Complete execution and assessment steps define the project work to build this agent-based architecture from beginning to end. This report explores the created system which utilizes pre-trained abilities of selected models.

---

2. Model Setup

- The text generation tasks inside agents used LLM: google/flan-t5-small which was selected among other models. It's accessed via langchain-huggingface.
- The movie overview embeddings are produced through the sentence-transformers/all-MiniLM-L6-v2 embedding system for semantic similarity engine access. The data structure became visible through a t-SNE visualization which processed the embeddings.
- The system employs langgraph (workflow) along with langchain (core abstractions) and transformers (pipeline) and sentence-transformers (embeddings) and pandas/numpy (data) and matplotlib (plotting) and tensorflow/tensorboard (logging) and graphviz (workflow visualization) as its main libraries.
- The system utilizes TMDB 5000 movie dataset which gets preprocessed to yield a collection of titles and genres along with overviews director data and vote averages and genres.

---

3. Implementation

Architecture: A hierarchical graph structure was built using langgraph.StateGraph. A master_router directs the workflow between specialized subgraphs (recommendation, personalization, explanation).

Agents/Tools: Key components include:

- movie_data_simulator: Retrieves candidate movies based on semantic similarity and genre filtering.
- context_parser: Interprets user context (e.g., "action-packed", "relaxed").
- explanation_formatter: Creates natural language justifications for recommendations.

State Management: A GraphState TypedDict manages information flow.

Code Structure: The code is well-organized with clear sections for setup, preprocessing, embedding visualization, agent/tool definitions, graph construction, evaluation (including TensorBoard logging), inference, and result visualization.

Workflow Visualization: The visualize_workflow function generates a styled and colored graph representing the agent interaction flow, enhancing clarity.

C**ommenting:** Comments explain the purpose of major code sections.

4. Model Evaluation, Results, and Findings

An extensive evaluation occurred through testing with predefined sets of test cases.

Evaluation Metrics & Procedures:

- **Core Performance:** Efficiency (response time), Robustness (success/fallback rate), Scalability (state transitions).
- **Relevance & Alignment:** Genre Relevance (match between query genre and recommendation genres), Context Alignment (match between context mood and recommendation genres).

- **Recommendation Quality:** The calculation considers two metrics: Diversity that represents director uniqueness in recommendations and Popularity that shows the average user rating level expressed as vote_average.
- **Visualization:** Visualization of metrics occurred through bar charts for each individual metric and one consolidated trend plot generated with matplotlib. The visualization of embedding space occurred through t-SNE implementation.
- **Logging:** Key metrics (Efficiency, Robustness, Scalability) were logged using TensorBoard for tracking.

Key Results: (Based on the 5 primary test cases)

- **Efficiency: Fast average response time of 0.129 seconds.**
- **Robustness: High success rate (4/5), correctly falling back on the invalid empty query**
- **Efficiency:** Fast average response time of 0.129 seconds.
- **Robustness:** The system achieved four out of five success rates while successfully transitioning to an empty query due to its invalidity.
- **Scalability:** Each successful run involved approximately 9.6 state transitions as the system operated efficiently.
- **Genre Relevance:** High average score of 0.92.
- **Context Alignment:** Perfect average score of 1.00 for cases with context.
- **Diversity:** The system proved effective in director recommendation by achieving an average score of 0.80 which matches the final result of 1.00 in all successful recommendations.
- **Popularity:** The calculations show a moderate popularity score of 4.73 based on vote_average values out of 10 throughout all test cases (including the baseline fallback score of 0). Recommendations that reached successful criteria averaged 5.9 on popularity score.

Findings & Observations:

- **Strengths:** The system functions effectively in most performance areas. The system accepts a wide range of query inputs and performs correctly with each type. The system

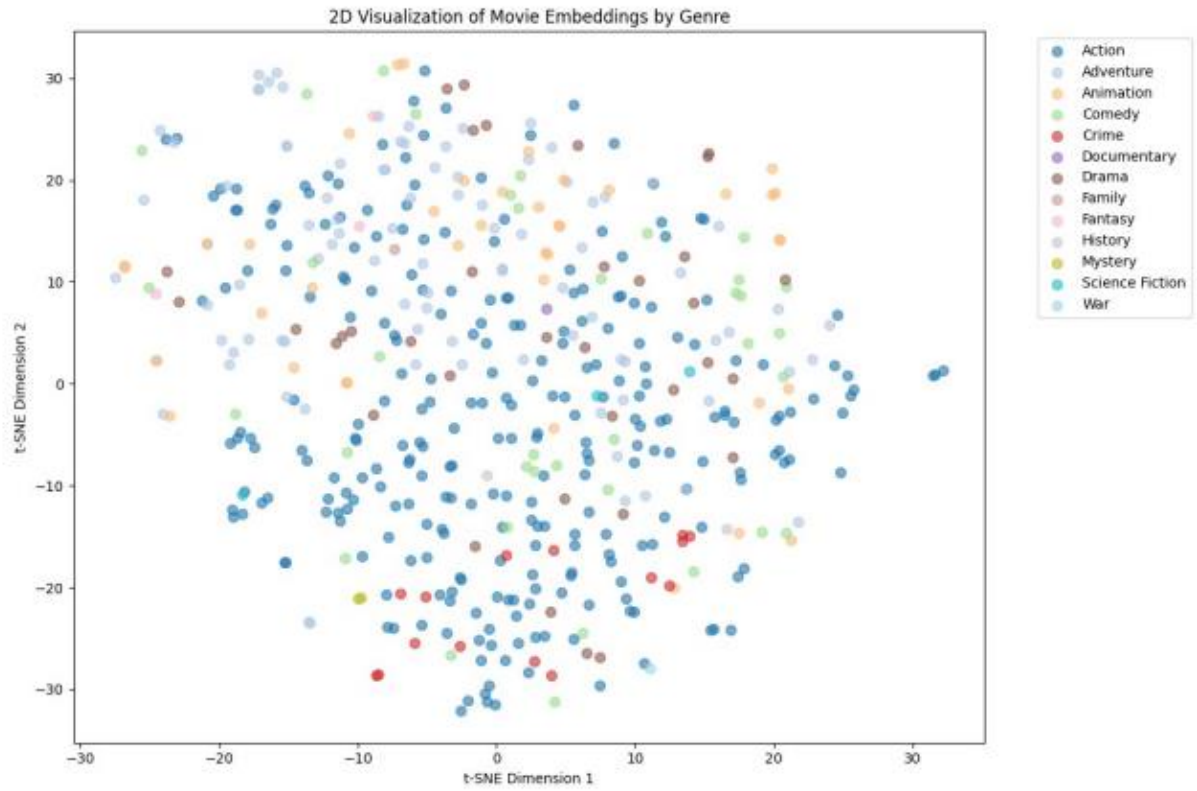
employs context sources while explaining its choices to generate suitable recommendations across a wide range of results. The system correctly operates the fall-back mechanism when users provide invalid input. Context alignment performs exceptionally well within this system. The diversity score demonstrates that the system does not present more than one film from the same director to users throughout a single suggestion session.

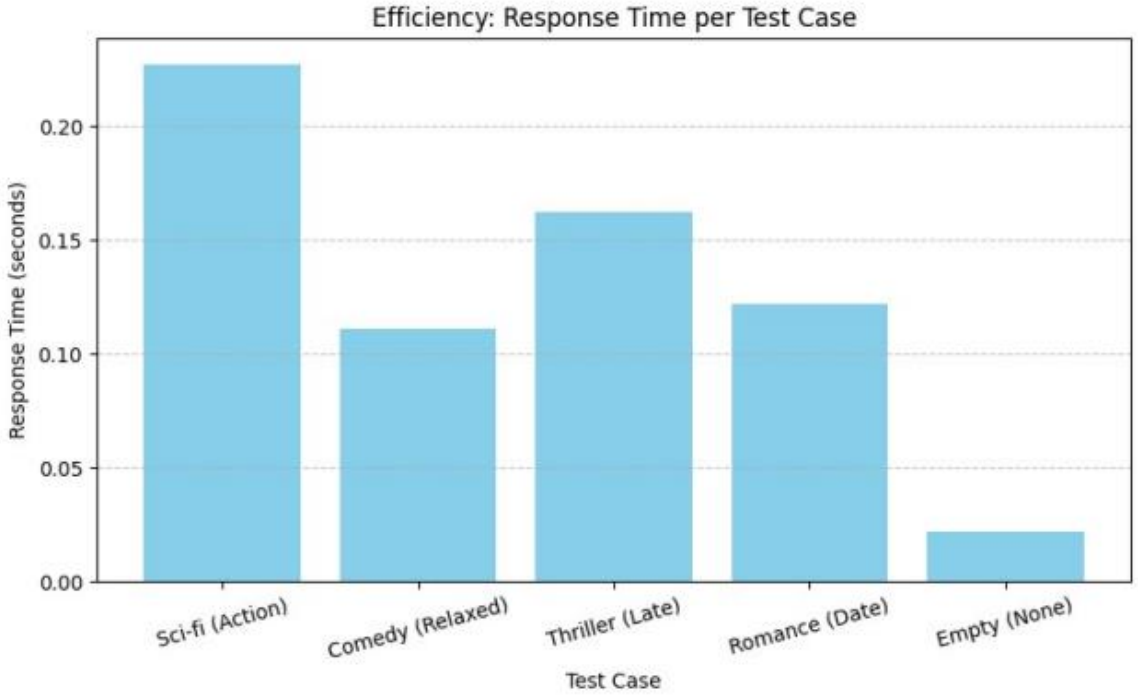
- **Areas for Refinement:** Rephrase the "sci-fi" search problem continues to exist where sci-fi recommendations combine with non-sci-fi action films even though the strong action context may need adjustment to improve the similarity search or filtering approach. The system demonstrates an average popularity score which indicates that it does not favor blockbusters although this rating might be regarded as too low or too high depending on user preferences.
- **Visual Insights:** The genre clusters become visible in the t-SNE plot although significant grouping between different categories demonstrates the difficulty of maintaining specific boundaries through overview embeddings analysis. Test case performance analysis becomes easy to understand through the consolidated performance trend plot which shows metric variations.

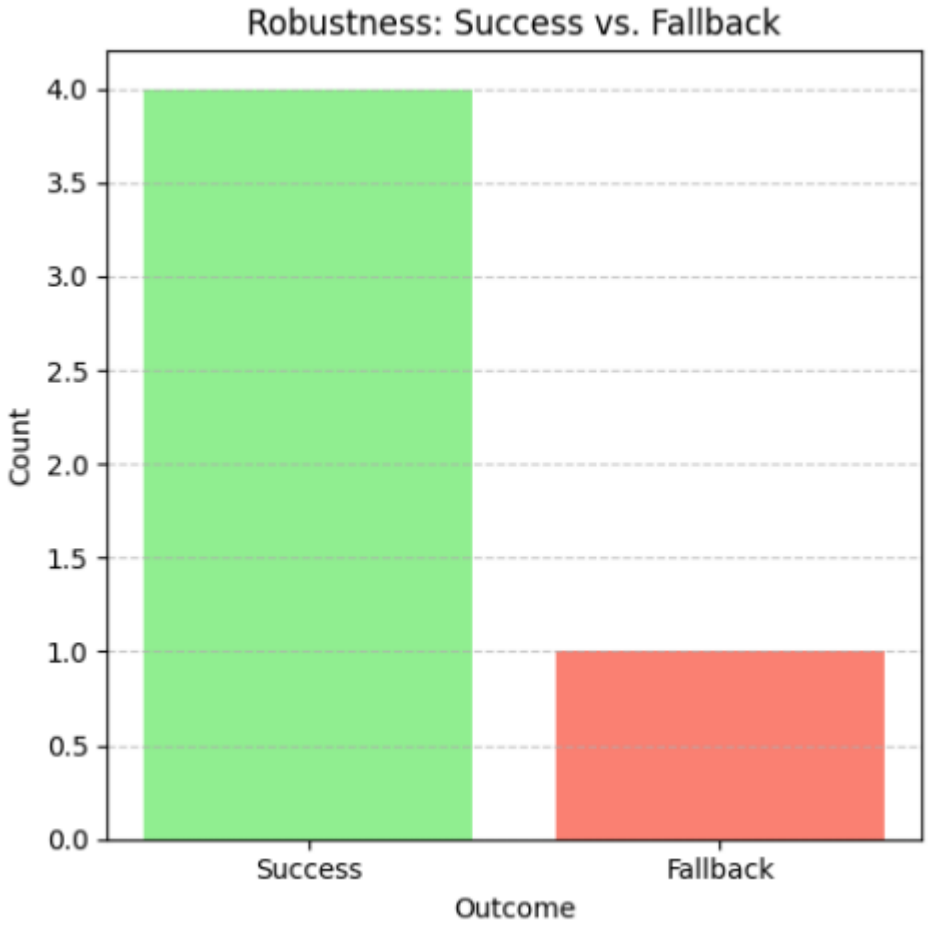
5. Visualizations

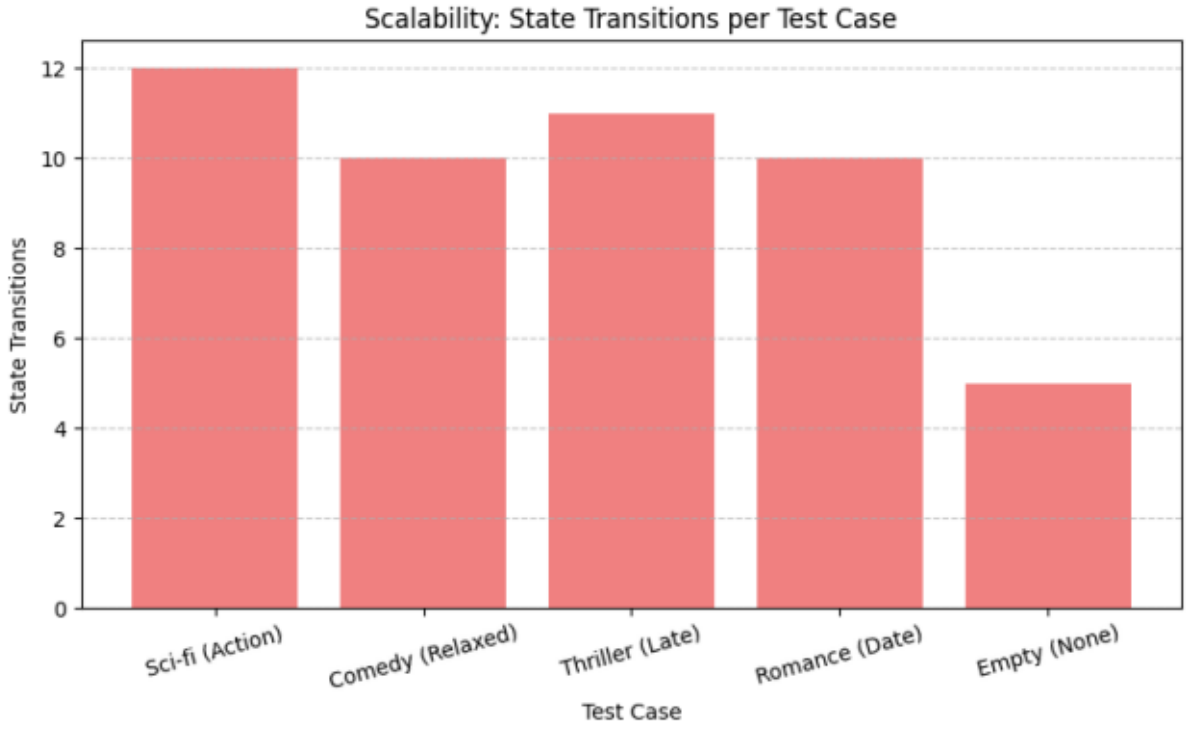
Multiple visualizations were generated to aid understanding:

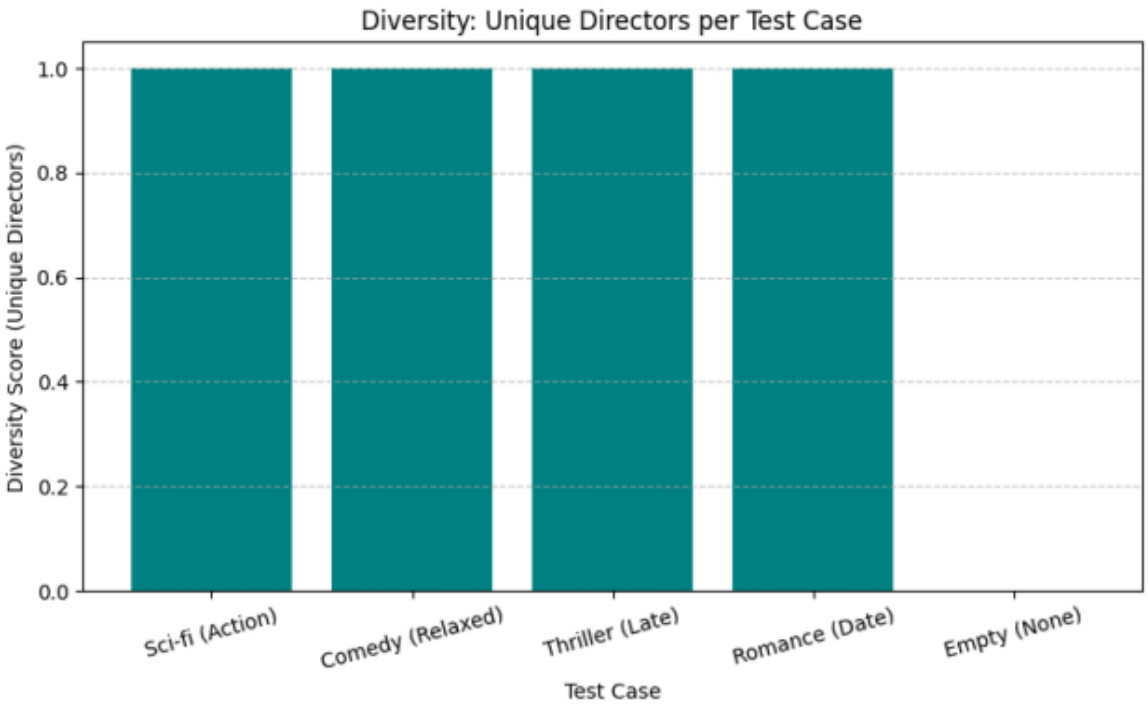
- t-SNE plot of movie embeddings by genre.
- Styled workflow graph showing agent interactions.
- Bar charts for Efficiency, Robustness, Scalability, Diversity, and Popularity per test case.
- Consolidated line plot showing trends of all key metrics across test cases ("Learning Curves").
- TensorBoard.

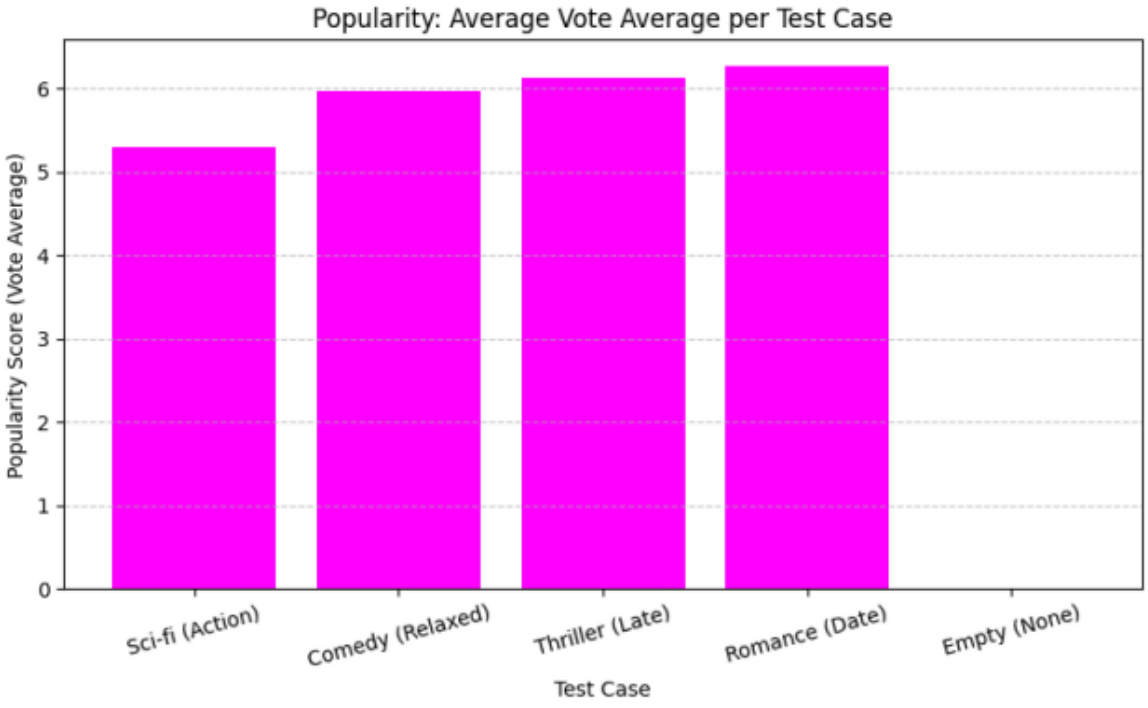

2D Visualization of Movie Embeddings by Genre
t-SNE Dimension 1
t-SNE Dimension 2
Action
Adventure
Animation
Comedy
Crime
Documentary
Drama
Family
Fantasy
History
Mystery
Science Fiction
War

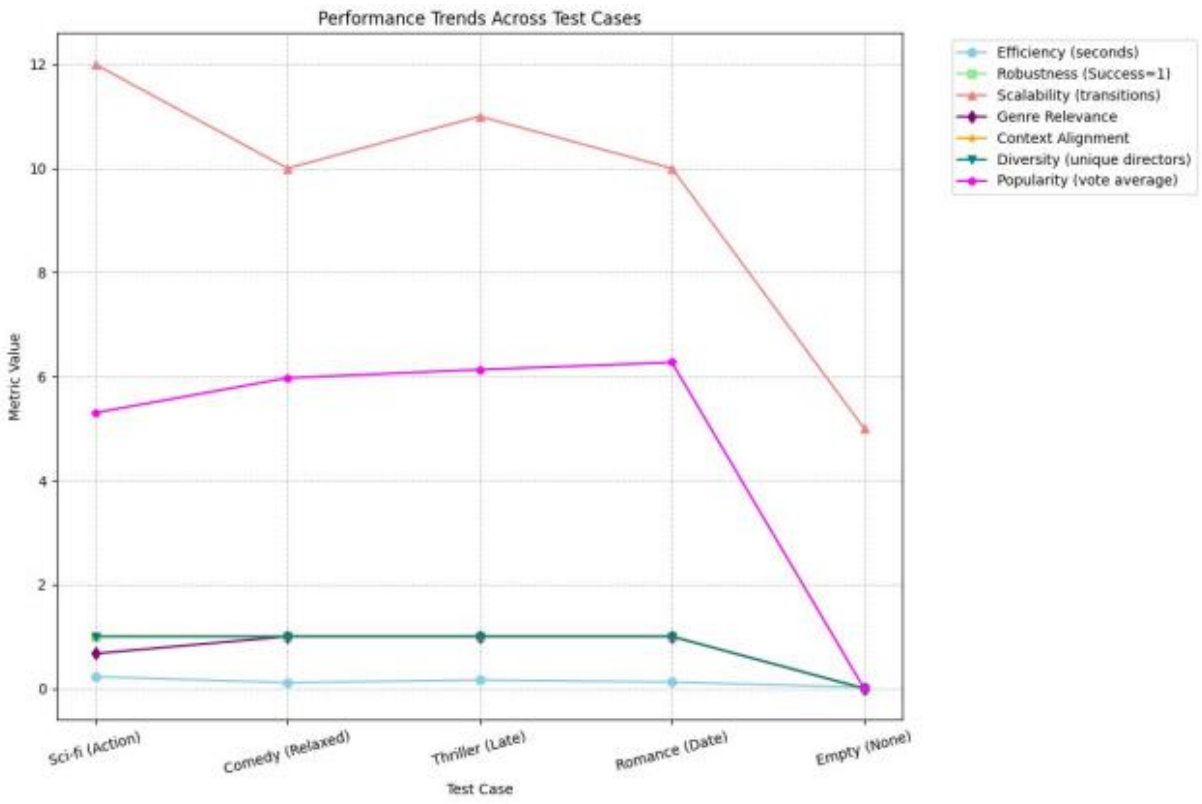

Efficiency: Response Time per Test Case
Response Time (seconds)
Test Case
Sci-fi (Action)
Comedy (Relaxed)
Thriller (Late)
Romance (Date)
Empty (None)


Robustness: Success vs. Fallback
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Count
Success
Fallback
Outcome


Scalability: State Transitions per Test Case
State Transitions
0
2
4
6
8
10
12
Sci-fi (Action)
Comedy (Relaxed)
Thriller (Late)
Romance (Date)
Empty (None)
Test Case


Diversity: Unique Directors per Test Case
Diversity Score (Unique Directors)
1.0
0.8
0.6
0.4
0.2
0.0
Sci-fi (Action)
Comedy (Relaxed)
Thriller (Late)
Romance (Date)
Empty (None)
Test Case


Popularity: Average Vote Average per Test Case
Popularity Score (Vote Average)
6
5
4
3
2
1
0
Sci-fi (Action)
Comedy (Relaxed)
Thriller (Late)
Romance (Date)
Empty (None)
Test Case



---

6. Conclusion

The system development and complete evaluation of a hierarchical multi-agent recommendation method for movies through LangGraph proved successful. Its architecture allows modular design together with contextual functions and explanation capabilities. The enhanced evaluation system features diversity and popularity metrics together with TensorBoard logging and t-SNE and workflow and performance plot visualizations which provide extensive understanding of the system behavior. Research shows that the system achieves high marks for performance efficiency and robustness as well as context matching abilities and diverse results generation. The system displays effective genre connection yet requires improvement in the filter mechanisms. The popularity evaluation balances favored blockbuster films and undiscovered and possibly unsolicited titles. Although the developers intentionally skipped fine-tuning this time they

demonstrated the transformation process for agent-based LLM applications by using TensorBoard and creating various performance metrics and plots.

---

8. References

[1] Peng, Q., Liu, H., Huang, H., Yang, Q., & Shao, M. (2025). *A Survey on LLM-powered Agents for Recommender Systems*. arXiv preprint arXiv:2502.10050..

[2] Zhu, K., Du, H., Hong, Z., Yang, X., Guo, S., Wang, Z., ... & You, J. (2025). *MultiAgentBench: Evaluating the Collaboration and Competition of LLM agents*. arXiv preprint arXiv:2503.01935.

[3] Xi, Z., Chen, W., Wang, X., He, H., Chen, Y., Feng, F., ... & Wang, X. (2024). *A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges*. arXiv preprint arXiv:2408.10651.

[4] Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP).

[5] Park, J., & Zhang, Y. (2025). *AgentRec: Agent Recommendation Using Sentence Embeddings Aligned to Human Feedback*. arXiv preprint arXiv:2501.13333.