LLM-Driven Conversational Movie Recommendation & Taste Graph Service

Emily Wang
Columbia University
Computer Science
New York, New York
eaw2233@columbia.edu

Eesun Moon
Columbia University
Computer Science
New York, New York
em3907@columbia.edu

Nikhil Sharma
Columbia University
Computer Science
New York, New York
ns3942@columbia.edu

***Abstract*—This document presents a movie recommendation system utilizing generative AI techniques. The system aims to provide personalized movie suggestions based on user preferences and viewing history.**

***Index Terms*—movie recommendation, machine learning, large language models, collaborative filtering, generative AI**

I. Clarity of Objectives

A. Clear Definition of Research Questions or Development Goals

Our development goals in this project are as follows:

- Construct a user's "taste embedding" representation
- Build an LLM-augmented conversational movie recommendation system
- Integrate lightweight personalization and explainability
- Implement a "Taste Wrapped" Visualization (Stretch goal)

Our first objective will be to encode each user's movie preferences by aggregating their ratings or dialogue interactions into a compact embedding vector that captures taste similarities. To do this, we plan on experimenting with using LLM sentence embeddings (e.g., text-embedding-3-large or bge-base) on movie metadata and user comments. We will also explore creating our own custom MCP server to wrap around TMDB and MovieLens data for providing the LLM "embedder" with additional context. Once we have generated these embeddings per user, we would like to illustrate them as a taste graph, allowing users to see how visually close they are to their friends and other users.

After we have developed the user's taste representation, our next goal would be to develop a conversational web interface that allows users to describe their movie preferences, favorite films, or desired themes (e.g., "a feel-good sci-fi movie under 2 hours"). This experience can utilize information from the user's "taste embedding" if the user has already provided it, or it can be completely stateless and just answer based on the conversation.

Going hand in hand with our second objective, our third objective is to demonstrate how LLM reasoning enhances transparency and trust compared to black-box recommendation systems (i.e. Netflix and other streaming services). Rather than just seeing a "We think you'll like this..." message, our goal is to be able to provide natural-language justifications such as "You might like Arrival because it shares the emotional tone and pacing of Interstellar." To evaluate this objective, we plan on testing improvement in explainability and user satisfaction via small-scale user testing or LLM-as-a-judge scoring.

As a stretch goal, we would like to build a summary report similar to Spotify Wrapped that highlights a user's yearly movie preferences and hidden themes discovered from embeddings. The primary challenge (and the reason we treat this as a stretch goal) for this objective would be to gather enough data from users in order to create this. Currently, popular movie recommendation websites such as Letterboxd and IMDB have limited API access for developers, so we're evaluating other mechanisms in which we can gather users' data with permission. We can always test this feature out with synthetic data as well, but would be really interested in being able to show users insights about their movie preferences they might not have known themselves.

B. Intended Methodology

1) System Overview: Our system will integrate three complementary components:

1) **Taste Embedding Generator** — constructs a latent vector representation of each user's movie preferences using both structured data (ratings, genres) and unstructured data (reviews, dialogue).
2) **LLM-Augmented Conversational Agent** — enables interactive, contextual preference elicitation and natural-language explanation.
3) **Visualization and Evaluation Modules** — provide interpretability through embedding visualization and narrative summary (Taste Wrapped).

The high-level pipeline is shown below:

- User Input
- Data Ingestion (TMDB/MovieLens/ReDial/CCPE)
- Embedding Generator
- Candidate Retrieval (FAISS/Pinecone)
- LLM Ranker + Explainer
- Output Recommendation
- Taste Wrapped Visualization

2) Data Sources and Preprocessing: We will combine both structured and conversational datasets:

- **MovieLens 1M/20M** [1]: baseline user–movie–rating matrix.
- **TMDB API** [14]: metadata (genres, plot, cast, keywords).
- **ReDial** [7] and **LLM-REDIAL** [6]: conversational recommendation dialogues.
- **CCPE (Coached Conversational Preference Elicitation)** [8]: annotated dialogues with explicit preference statements.
- **GoEmotions** [15]: fine-tuning dataset to train the model with emotional understanding.
- **MovieTweetings** [16]: movie ratings based on Twitter (recent and informal reviews).
- **Netflix Prize Dataset** [17]: 480K users, 17K movies, 100M rating records.
- **INSPIRED** [18]: recommendation dialogues with emotional intent annotations.
- **Synthetic Augmentation**: generate controlled dialogues using GPT-4 or similar models to increase coverage and variety.

Movies and user comments will be encoded while testing recent embedding models alongside a few classic ones, to compare runtime and performance: :

- **text-embedding-3-large** (OpenAI 2024)
- **bge-base-en v1.5** [Xiao et al., 2023]
- **GTE-base / Jina-embeddings** (2024), known for multi-domain semantic performance.

*3) **Conversational Recommendation and Ranking**:* The **Conversational Recommender** module follows a retrieval-augmented generation (RAG) design:

1) Retrieve top-K similar movies from the embedding index using cosine similarity.
2) Pass retrieved items and user context into an **instruction-tuned LLM ranker**, based on the *RecRanker* [4] or *ChatRec* [5] paradigm.
3) The LLM outputs both ranked recommendations and **rationales**.

We will compare two variants:

- **RAG-Only Baseline:** static embedding retrieval with template explanations.
- **LLM-as-Ranker Model:** fine-tuned or prompted model performing joint ranking + explanation (RecRanker-style).

Evaluation metrics will include Recall@K, explanation helpfulness, and user satisfaction.

4) Explainability: To address transparency, we implement an *LLM-Explainer* that uses **Chain-of-Thought prompting** [12] to reason about similarities and generate justifications:

"Given user taste vector and retrieved movies, reason step-by-step about tone, pacing, and theme overlap, then summarize in one sentence."

Explanations will be evaluated using the framework from Jiang et al. (2024) [9], measuring factual accuracy, contextual consistency, and diversity of reasoning. We will also conduct a small user study to judge our process.

5) Stretch Goal: Visualization: For the stretch goal, we will use **UMAP or t-SNE** to project user and movie embeddings to 2-D space. Each user's trajectory over time can be annotated by genre or emotion clusters. We will employ **LLM-narrative generation** [11] to produce personalized summaries such as:

"In 2025, your movie choices shifted from high-intensity thrillers to character-driven dramas."

This visualization serves both as an interpretability feature and a user engagement tool.

C. Expected Outcomes and Deliverables

Our primary expected outcome is a functional LLM-augmented conversational movie recommendation system that demonstrates how semantic taste embeddings can enable personalized, interpretable, and conversational recommendations. The quality of recommendations will serve as a proxy for the representational quality of user taste embeddings.

We anticipate three core deliverables:

1) **Taste Embedding Framework**: a pipeline that effectively maps users and movies into a shared embedding space, enabling similarity-based retrieval and visualization.
2) **Conversational Recommender Interface**: a chat-based system capable of interactive movie recommendation and natural-language explanation generation.
3) **Stretch Goal - Explainability Visualization ("Taste Wrapped")**: an interpretable report visualizing each user's preference trajectory and highlighting embedding clusters that capture genre, tone, and theme similarity.

D. Alignment with Course Objectives

This project directly aligns with multiple core topics from the course syllabus, particularly LLM application development, Retrieval-Augmented Generation systems, and efficient serving techniques. Our conversational recommendation system applies prompt engineering and RAG concepts to create a practical GenAI application that addresses real-world limitations of LLMs, such as knowledge cutoffs and hallucinations, by grounding recommendations in structured movie databases. Additionally, the project incorporates tool-assisted LLM concepts through our custom MCP server integration with TMDB and MovieLens APIs, and explores multi-agent patterns through the modular design of our taste embedding generator, conversational agent, and explanation modules.

II. Feasibility

A. Realistic Timeline and Milestones

The proposed project is feasible within the rest of the semester, built on top of existing datasets, open-source models, and accessible APIs. Most pipeline components — embedding generation, retrieval, and LLM prompting — can be implemented using off-the-shelf libraries (HuggingFace Transformers, FAISS, LangChain, Streamlit). Timeline - approximately 1 week per milestone Literature review, data collection

(MovieLens, TMDB, ReDial, CCPE) Taste embedding construction, vector database setup Conversational recommendation interface (RAG + LLM-as-ranker) Explainability module (LLM explanations, evaluation prompts) (stretch goal) Taste Wrapped visualization and evaluation design User study and final integration

B. Assessment of Required Resources and Their Availability

Computing: Development on local compute and GPU cloud (GCP credits and Colab) APIs: TMDB, OpenAI embeddings API, or open-source equivalents (BGE, Jina, Instructor embeddings). Frameworks: PyTorch, FAISS/Pinecone, LangChain for RAG; Streamlit or FastAPI for web interface. Collaboration Tools: GitHub for version control, Weights & Biases for tracking experiments, and Google Drive for dataset caching.

C. Identification of Potential Challenges and Mitigation Strategies

Several potential challenges are anticipated in developing the proposed system, along with corresponding mitigation strategies, as outlined below.

- **Data availability:** Certain APIs such as TMDB may impose rate limits, restricting large-scale data access. To mitigate this issue, we plan to rely on publicly available open datasets such as MovieLens, ReDial, and CCPE. In cases where these datasets do not provide sufficient coverage, synthetic data generation using GPT-based models or web scraping from publicly available sources can serve as fallback options.
- **Limited user information:** Public datasets that include detailed user activity or preference histories for movie recommendations are often scarce. To address this limitation, we will adopt the same mitigation approaches as in the previous point, while also leveraging preference dialogue training on the LLM side to generalize user intent understanding beyond specific movie-based fine-tuning data.
- **Computation cost:** The computational cost of training or fine-tuning large models may become prohibitive. If this issue arises, we will mitigate it by utilizing more compact or quantized LLMs, as well as efficient fine-tuning methods such as LoRA or adapter-based optimization.
- **Evaluation difficulty:** Assessing subjective or qualitative aspects of conversational recommendation poses inherent challenges. To address this, we will employ the LLM-as-a-Judge [13] framework, alongside evaluation panels composed of diverse LLMs, to provide a more reliable and multi-perspective qualitative assessment.

D. Clear Scope Definition

The scope of this project is to develop a conversational and explainable movie recommendation system centered on individual user preferences. Rather than implementing user-to-user matching or social recommendation features, the focus will remain on personalized retrieval and reasoning. The system aims to encode each user's unique "taste embedding" derived from ratings, metadata, and dialogue interactions, which will serve as the foundation for recommendation and natural-language explanation.

The implementation will include the following major components: (1) a **Taste Embedding Generator** that constructs user embeddings using datasets such as MovieLens, TMDB, and ReDial; (2) a **Conversational Recommender** that elicits preferences and generates recommendations via retrieval-augmented generation (RAG) with LLM-as-ranker reasoning; (3) an **Explainability Module** that produces human-interpretable rationales for recommendations; and (4) a **Taste Wrapped Visualization** (stretch goal) that presents users' evolving preferences through 2-D embedding projection and narrative summarization.

The following aspects are explicitly out of scope for this semester: (i) user-user matching or community-based recommendation; (ii) reimplementation of collaborative filtering or matrix factorization algorithms; (iii) full-scale backend infrastructure or deployment; (iv) real-user data logging or privacy-compliance mechanisms; and (v) multimodal extensions involving visual or audio features.

By constraining the scope to a prototype-level, individual-centered system, the project remains feasible within the semester timeline while contributing novel insights into how LLM reasoning and semantic embeddings can jointly enhance personalization and interpretability in recommendation systems.

III. Innovation and Relevance

A. Originality of the Proposed Approach

This project advances beyond conventional recommender pipelines by:

- **Integrating LLM reasoning into both retrieval and explanation stages**, not merely as a dialogue layer.
  - Introducing **"taste embeddings"** derived from multimodal data (ratings, comments, metadata).
  - Combining **semantic embeddings + graph-aware reasoning** with conversational preference elicitation.
  - Generating **narrative-based visual explanations** (Taste Wrapped) — a hybrid of recommender transparency and narrative generation.

The integration of *LLM-as-ranker* [4] and *Chain-of-Thought explanation* [12] represents a novel methodology that connects GenAI reasoning to recommendation interpretability.

B. Connection to Current GenAI Research Trends

- This work builds on a growing body of literature exploring **LLM4Rec** [3], **LLM-Enhanced Recommender Systems** [2], and **Conversational Explainability** [10]. It contributes to two emerging trends:
- **Human-AI collaboration** — systems that elicit preferences conversationally instead of inferring them implicitly.
- **Transparent personalization** — bridging black-box recommendations and interpretable narrative reasoning.

- The project thereby aligns with the current GenAI emphasis on *trustworthy, explainable, and user-centric* systems.

C. Potential Impact on the Field

If successful, this system would demonstrate how semantic embeddings and LLM reasoning can produce explainable, interactive recommenders without requiring massive proprietary datasets. Furthermore, the success of our ”LLM embedder” approach would provide additional evidence for the capability of large language models to understand and capture complex human sentiments and preferences in nuanced, contextually-aware representations.

Potential impacts include:

- Enabling research reproducibility through open data and models
- Inspiring hybrid architectures that combine retrieval and reasoning for other domains such as music, books, and games
- Providing a template for GenAI-based transparency in personalized systems
- Offering educational value for understanding interpretability in human-AI interfaces

D. Novel Application or Methodology

A key methodological innovation in our approach is the development of a custom MCP server that wraps around TMDB and MovieLens data sources, enabling our LLM model to dynamically access and reason over movie metadata with richer context than traditional static databases. Rather than relying solely on pre-computed embeddings, our ”LLM embedder” approach allows the system to generate contextually-aware taste representations on-the-fly by processing user comments, ratings, and conversational inputs through modern sentence embedding models (e.g., text-embedding-3-large, bge-base). By combining MCP-based dynamic data access with LLM-powered embedding generation, our system bridges the gap between structured recommendation pipelines and natural language understanding, creating a more adaptive and interpretable recommendation experience.

REFERENCES

[1] F. M. Harper and J. A. Konstan, “The MovieLens Datasets: History and Context,” *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2015.

[2] Q. Liu, W. Zhang, C. Xu, and X. He, “LLM-Enhanced Recommender Systems (LLMERS): A Survey,” *arXiv preprint arXiv:2412.13432*, 2024.

[3] H. Gao, Y. Wang, R. Zhang, and T.-S. Chua, “LLM4Rec: Large Language Models for Recommendation,” *arXiv preprint arXiv:2401.08350*, 2024.

[4] S. Luo, B. He, H. Zhao, W. Shao, Y. Qi, Y. Huang, A. Zhou, Y. Yao, Z. Li, Y. Xiao, M. Zhan, and L. Song, “RecRanker: Instruction-Tuning Large Language Model as Ranker for Top-k Recommendation,” *arXiv preprint arXiv:2312.16018*, 2023.

[5] Y. Gao, T. Sheng, Y. Xiang, Y. Xiong, H. Wang, and J. Zhang, “Chat-REC: Towards Interactive and Explainable LLMs-Augmented Recommender System,” *arXiv preprint arXiv:2303.14524*, 2023.

[6] X. Liang, Y. Chen, Y. Tang, and X. Li, “LLM-REDIAL: A Large-Scale Dataset for Conversational Recommender Systems,” in *Findings of the Association for Computational Linguistics (ACL Findings)*, 2024.

[7] R. Li, S. E. Kahou, H. Schulz, V. Michalski, L. Charlin, and C. Pal, “ReDial: Recommendation Dialogues Dataset,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

[8] F. Radlinski, K. Balog, B. Byrne, and K. Krishnamoorthi, “Coached Conversational Preference Elicitation: A Case Study in Understanding Movie Preferences,” in *Proceedings of the 20th SIGDIAL*, 2019.

[9] Y. Jiang, Y. Wang, and W. Xin Zhao, “Beyond Utility: Evaluating LLM-based Recommenders,” *arXiv preprint arXiv:2411.00331*, 2024.

[10] A. Said, K. Verbert, and F. Ricci, “On Explaining Recommendations with Large Language Models,” *Frontiers in Big Data*, 2025.

[11] Spotify Research, “Contextualized Recommendations Through Personalized Narratives Using LLMs,” 2024.

[12] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chen, Q. Le, D. Zhou, and E. Chi, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[13] H. Huang, X. Bu, H. Zhou, Y. Qu, J. Liu, M. Yang, B. Xu, and T. Zhao, “An Empirical Study of LLM-as-a-Judge for LLM Evaluation: Fine-tuned Judge Model is not a General Substitute for GPT-4,” *arXiv preprint arXiv:2403.02839*, 2024.

[14] The Movie Database (TMDB), “TMDB API: The Movie Database Developer Platform,” *Available at:* https://developer.themoviedb.org/docs/getting-started, Accessed: 2025.

[15] D. Demszky, D. Movshovitz-Attias, S. Ko, A. Cowen, G. Nemade, and S. Agrawal, “GoEmotions: A Dataset of Fine-Grained Emotions,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[16] T. Dooms, “MovieTweetings: A Movie Rating Dataset Collected From Twitter,” *GitHub Repository*, 2013. [Online]. Available: https://github.com/sidooms/MovieTweetings

[17] Netflix Inc., “Netflix Prize Dataset,” *Kaggle Dataset*, 2009. [Online]. Available: https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data/data

[18] R. Lubis, M. Cercas Curry, and E. Casanueva, “INSPIRED: A Dataset for Emotion-Aware Conversational Recommendation,” in *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, 2019. [Online]. Available: https://github.com/sweetpeach/Inspired