IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 27, Issue 6, Ser. 3 (Nov. – Dec. 2025), PP 01-10
www.iosrjournals.org

A Hybrid Content Recommendation System Leveraging Markov Chains, Personalised Similarity, and Large Language Models for Next-Item Prediction

Spandan Pratyush
Independent Researcher, Delhi, India

Abstract
In the age of content consumption by millions of users and tracking of consumed data, having a good recommendation system is still evolving strategically. This has been a highly researched topic, and over the years, the most common methods are based on Collaborative Filtering for finding user-based similarity and Content-based similarity[1]. For finding the best correlations, cosine similarity, Pearson coefficient similarity, Jaccard similarity, etc,[1,2] have been researched. The limitations are still related to user behaviour mapping. Users can have a wide range of interests or can be very domain specific, and this entire information can't be encoded just in user similarity, but some of it is also encoded in the content. This does not necessarily mean the transition from one content state to another state based on content similarity. This method tries to bring the relevance of the age-old mechanism of Markov Chains to get the transitional information. Markov Chains have been time-tested in finding the best transitions by their usage in the Google PageRank algorithm[3]. By maintaining the ranking of transitions from pages, the search engine has maintained the relevance of search results based on actual transitions. The core method of this approach is to bring back those transitional probabilities as an important metric in recommendations of relevance. The problem can be defined by a state of the content consumed (watched, book read, etc) for a particular user and for which we fetch recommendations for the next content based on a hybrid approach. First is a Markov Chain to get the ranks of content based on transitional probabilities. Alongside this, to encode the particular user behaviour, the user-based similarity is maintained by calculating Jaccard similarity scores, as this encodes the similarity of two sets. The idea behind this is that the most similar users will have similar transitions. From the sorted Jaccard scores, top_k similar users who have the particular content in their watch history, the linked content to those (previous and next) are also fetched. The third step is a simple LLM-based rerank of both retrieved sets. This similarity is the content metadata collaborative filtering using Cosine Similarity. The overall aim is to add content transitions as an important metric for finding recommendations for a given state, as transitions encode the content-based behaviour in the most relevant way.

Date of Submission: 25-11-2025 Date of Acceptance: 08-12-2025

I. INTRODUCTION

Recommendation systems play an important role in most modern digital products. They are already multiple applications in Ecommerce products (user personalisation), content streaming (videos or movies), book reviews and recommendations, media (article recommendation) etc. Apart from collaborative filtering by content and user and development of personalised models in various spaces by the use of user profile and calculating relevance scores. With increasing data, **matrix factorization** and **deep learning (convolutional or recurrent neural networks or autoencoders)** have become more relevant. The limitations of these have primarily been that the methods have been domain specific. The systems also fail to read contextual information from multimodal data [4]. This is where LLM based approaches are being researched feeding user-item interactions and all multimodal data to provide recommendations [5] requiring prompt engineering and Low Rank Adaptation (LoRA) based fine tuning for the best results. This approach tries to reduce the burden of capturing multimodal information of all content by any deep learning or LLM based approach by deploying a hybrid approach of using Markov chains and Collaborative User Filtering based transitions to encode that information. The method defines the problem statement as providing the options of next content based on the the context Cx consumed by user U. The inspiration of the approach is derived from the design of search engines. The most famous algorithm for search engines has been Google's PageRank algorithm which is based on Markov Chains [3]. The reason why the search algorithms have always provided relevant results because they actually store the information of user behavior among connected content having relationship with each other. Markov chains were developed by Russian mathematician **Andrei A. Markov** (1856-1922) in an attempt to prove the dependency within a system whose theorisation was philosophical in the way our world works. We model tne transition from one state to another on

the basis of transitional probabilities. [6] Markov had worked on the transitional probabilities of vowels and consonants within language and it can applied to systems where there are distinct states with transitions between them.[7] The user behaviour on the internet is defined by transitions and thus the Google PageRank algorithm has been able to capture it. The user activity on content consumption platforms can be defined similar to how users have searched on the web and since the number of content on a ny platform is limited, it is a closed system where transitions will have probability. The only challenge to this arises from types of user behaviour which thus brings the second aspect of the approach which is personalization. While some users can be limited in the genre or other attributes of content consumption, some users might be explorers. This adds the need for user based collaborative finding to find the most similar users [1,2]. So, additionally we can also check for transitions or links between our candidate content Cx in the filtered similar users list with the next or previous content to find relevant content specific the the user. While the Markov transitions focus more on relevance as a natural behaviour, **user collaborative filtering** adds user based novelty to the set of recommendations. With this filtered list, LLM based reranking on the content metadata for similarity. This uses cosine similarity of content based filtering to rank as researched extensively for recommendations. [8, 9, 10]. This hybrid approach tries to balance relevance with diversity and novelty for content and can be the foundational filter for all kinds of content to be added to a variety of other business approaches like a mix of newly added content, a higher relevance score to user activity of liking a certain content etc. It primarily tries to add the information encoded in user transitions to recommendations.

II. RELATED WORK

The landscape of content recommendation systems is vast and continually evolving, driven by the increasing volume of available content and the diverse preferences of users. Proposed hybrid system draws inspiration from several established paradigms: sequential recommendation, collaborative filtering, hybrid architectures, and the emerging field of Large Language Models (LLMs) in recommendation.

2.1. Sequential Recommendation Systems and Markov Chains

Next-item prediction, a core task in recommendation, focuses on suggesting items that a user is likely to interact with next, given their recent sequence of actions. Early approaches to sequential recommendation often leveraged Markov Chains (MCs) due to their simplicity and ability to model transitions between states (items). A first-order Markov Chain assumes that the probability of the next item depends only on the current item, capturing immediate dependencies in user behavior [26]. Higher-order Markov Chains extend this by considering a longer history of items, allowing for more complex sequential patterns.

A significant advancement in this area was the **Factorized Markov Chains (FPMC)** model proposed by [24]. FPMC combines the concept of Markov Chains with matrix factorization techniques, allowing for a more robust handling of sparse interaction data and the cold-start problem for items. Subsequent works explored variations and extensions, such as personalized Markov Chains (Liu et al., 2011) and those incorporating temporal dynamics. While Markov Chains are computationally efficient and highly interpretable, their inherent "memoryless" property (or limited memory in higher-order variants) can restrict their ability to capture long-range dependencies or complex contextual factors that influence user choices.

With the rise of deep learning, more sophisticated sequential models emerged. Recurrent Neural Networks (RNNs), particularly Gated Recurrent Units (GRUs) and Long Short-Term Memory (LSTMs), have been successfully applied to session-based recommendation [18]. These models can theoretically capture longer-term dependencies than traditional Markov Chains. More recently, Transformer-based architectures have demonstrated state-of-the-art performance in sequential recommendation. Models like **SASRec (Self-Attentive Sequential Recommendation)** [19] leverage self-attention mechanisms to weigh the importance of different past items in predicting the next one, effectively capturing relevant dependencies regardless of their position in the sequence. Similarly, **BERT4Rec** [28] adapts the bidirectional encoder representations from Transformers to model user behavior sequences, further enhancing contextual understanding. While powerful, these deep learning models often require substantial computational resources and large datasets to train effectively, and their interpretability can be lower than that of simpler Markov Chains. Our approach re-examines the utility of Markov Chains for their efficiency and interpretability in generating an initial candidate set, before leveraging more advanced techniques.

2.2. Collaborative Filtering and User Similarity

Collaborative Filtering (CF) is a cornerstone of recommendation systems, operating on the principle that users who agreed in the past will agree again in the future [20, 25]. This paradigm broadly divides into user-based CF and item-based CF. **User-based CF**, highly relevant to our "Personalization Similarity" module, identifies a group of users whose past preferences or behaviors are similar to the current user's. Recommendations are then generated by aggregating the items preferred by these "neighboring" users. Common similarity metrics

include Jaccard similarity (for set-based overlaps), Cosine similarity (for vector-based representations of item interactions), and Pearson correlation coefficient.

The strength of user-based CF lies in its ability to discover novel and diverse items (serendipity) that the current user might not find through content-based methods alone, as it taps into the collective intelligence of the community. However, it can suffer from data sparsity, especially for users with limited interaction history (the "cold-start" user problem), and its performance can degrade in very large datasets due to the computational cost of finding neighbors. Our method specifically applies user similarity in a focused context, identifying users similar to the current user *around a specific item interaction (cx)*, thereby grounding the personalization in a relevant behavioral context.

2.3. Hybrid Recommendation Systems

The limitations inherent in purely content-based or collaborative filtering approaches often motivate the development of **hybrid recommendation systems**, which combine multiple techniques to leverage their respective strengths and mitigate their weaknesses. The hybrid approaches [13, 14] provide a comprehensive taxonomy of hybrid approaches, including weighted, switching, mixed, feature combination, cascade, and meta-level hybridization.

Hybrid systems have been shown to often outperform single-paradigm models by addressing issues like cold-start problems, sparsity, and over-specialization. Examples include combining collaborative filtering with content-based filtering [22] or integrating matrix factorization with contextual information [11]. Our proposed system falls under the hybrid category, specifically combining a global sequential model (Markov Chains) with a personalized collaborative component (user similarity based on contextual activity) and a semantic re-ranking layer (LLM). This multi-stage approach aims to build a robust candidate set before refining it with advanced semantic understanding.

2.4. Large Language Models (LLMs) in Recommendation

The recent advancements in **Large Language Models (LLMs)**, such as GPT [12], BERT [15], and their successors, have opened new avenues for recommendation systems. LLMs possess remarkable capabilities in understanding, generating, and processing human language, which can be leveraged to interpret rich item metadata (titles, descriptions, tags) and user reviews.

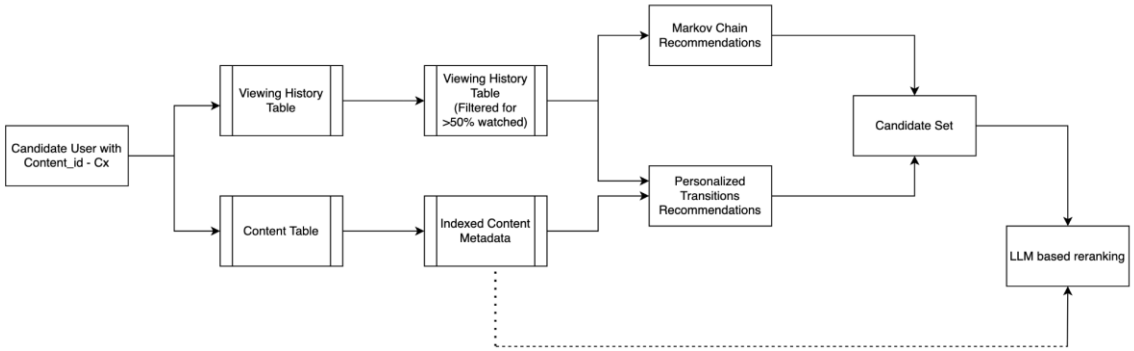
Initial applications involved using pre-trained language models to generate **item embeddings** or **user embeddings** that capture semantic relationships, which could then be used for similarity-based recommendations (e.g., embedding items and users into a shared space for nearest-neighbor search) [17]. More sophisticated approaches have explored **LLMs as generative recommenders**, capable of directly generating recommendation lists or personalized explanations for recommendations, often in a conversational context [5]. Crucially for our work, LLMs are also being employed as powerful **re-rankers** in multi-stage recommendation pipelines. After an initial set of candidates is generated by traditional methods, an LLM can evaluate these candidates based on their semantic coherence with the user's past interactions, the query item, and other contextual information (e.g., using prompts to ask the LLM to rank items based on specific criteria). This leverages the LLM's ability to reason about complex textual features and nuanced relationships that purely statistical models might miss. While LLMs offer unprecedented semantic understanding, their deployment in recommendation systems faces challenges related to computational cost, inference latency, potential for hallucination, and the need for careful prompt engineering [27]. Our system integrates an LLM specifically as a final re-ranking layer, aiming to harness its semantic capabilities for fine-grained contextual relevance while mitigating some of the computational overhead by operating on a pre-filtered candidate set.

In summary, our research synthesizes the established efficacy of Markov Chains for capturing sequential patterns, the personalized insights from user-based collaborative filtering, and the advanced semantic reasoning capabilities of LLMs within a novel hybrid architecture designed for next-item content prediction. This approach addresses the limitations of individual paradigms by providing a robust, personalized, and semantically informed recommendation strategy.

III. PROPOSED METHODOLOGY

This section outlines the design and implementation of the proposed hybrid content recommendation system, which leverages the complementary strengths of Markov Chains, personalized user similarity, and Large Language Models (LLMs). The overall architecture, depicted in Figure 1, comprises three main stages: Data Preprocessing, Candidate Generation, and LLM-Enhanced Re-ranking. Given a user who has just watched a specific content item (cx), the system aims to provide a personalized and semantically relevant list of recommended k content items.

Figure 1: Hybrid Content Recommendation System Architecture



3.1. Data Preprocessing

The system's foundation is built upon two primary datasets:

1. **User Viewing History Table:** Contains records of user interactions, including user_id, content_id, watch_timestamp, and watch_duration_percentage.
2. **Content Metadata Table:** Provides descriptive attributes for each content_id, such as title, genre, description, and tags.

The preprocessing pipeline (as shown in Figure 1, "Data Preprocessing" stage) involves the following steps:

- **Filtering by Watch Duration:** To ensure genuine user interest and filter out accidental clicks or brief previews, only viewing events where watch_duration_percentage is at least 50% are retained. This threshold helps in capturing meaningful engagement patterns.
- **User Sequence Construction:** For each unique user_id, their filtered viewing history records are ordered chronologically by watch_timestamp to form individual viewing sequences. These sequences represent the path a user takes through the content catalog.
- **Content Metadata Indexing:** The content_metadata is loaded into an in-memory dictionary or similar data structure, allowing for efficient lookup of title, genre, description, and tags by content_id. This metadata is crucial for the LLM-enhanced re-ranking module.

3.2. Candidate Generation

This stage is responsible for generating an initial, diverse set of potential next-item recommendations. It operates in parallel, combining global sequential patterns with personalized user behavior.

3.2.1. Markov Chain-based Global Recommendations

This module captures general, aggregated sequential viewing patterns across all users. It is based on a first-order Markov Chain model, where the probability of watching a particular content item depends only on the immediately preceding item.

- **Transition Matrix Construction:**

The system iterates through all constructed user_viewing_sequences in the dataset. For every consecutive pair of content items (C_i, C_j) in a user's sequence (where C_j immediately follows C_i), a counter count (C_i -> C_j) is incremented. After processing all sequences, these counts are converted into conditional probabilities:

$$P(C_j|C_i) = \frac{\text{count}(C_i \to C_j)}{\sum_k \text{count}(C_i \to C_k)}$$

This results in a transition matrix where each entry $P(C_j \mid C_i)$ represents the probability that a user will watch content C_j given they just watched content C_i

- **Candidate Retrieval:** Given the current content cx, the module queries this transition matrix to identify all content items C_j for which $P(C_j \mid cx) > 0$. The *top_k_mc* items with the highest probabilities are selected as Markov Chain candidates. These candidates reflect prevalent global trends in content consumption following cx.

3.2.2. Personalization Similarity Module

To inject user-specific relevance beyond global trends, this module identifies content preferences from users who exhibit similar prior behavior to the current user around the content cx.

- **User Similarity Metric:** For a given current_user_id and cx, similarity is calculated between current_user_id and other users who have also watched cx. The similarity metric used is the Jaccard Similarity of the sets of content items watched *before* cx by both users.

Let SU_{curr,pre_cx} be the set of content items watched by the current user (U_{curr}) prior to cx, and SU_{other,pre_cx} be the corresponding set for another user (U_{other}). The similarity is calculated as:

$$Jaccard\ Similarity(U_{curr}, U_{other}) = \frac{|SU_{curr,pre\_cx} \cap SU_{other,pre\_cx}|}{|SU_{curr,pre\_cx} \cup SU_{other,pre\_cx}|}$$

This metric focuses on shared taste leading up to a common interaction point, making it highly contextual.

- **Identifying Similar Users:**

1. The system identifies all users in the dataset who have also watched cx.
2. For each such user, their Jaccard similarity with current_user_id is computed based on their viewing histories *preceding* cx.
3. The top_k_sim_users with the highest similarity scores are selected.

- **Candidate Retrieval:** For each of the top_k_sim_users, the content item they watched immediately before and after cx in their respective sequences is identified and collected. These items form the set of personalized candidates. This component caters to users who might have idiosyncratic tastes or explore content in less common but coherent ways, reflecting similar users' journeys.

3.2.3. Candidate Set Generation

The candidates from both the Markov Chain module and the Personalization Similarity module are combined to form the Initial Candidate Set. This is done by taking the **union** of the two sets of content IDs. Crucially, the content item cx itself is removed from this set to prevent recommending an item the user has just finished watching. If, after this process, the candidate set is empty (e.g., cx is a very rare item or at the end of many sequences), a fallback strategy can be employed, such as recommending globally popular items or a random selection from the catalog to maintain system robustness.

3.3. LLM-Enhanced Re-ranking

The Initial Candidate Set from the previous stage, along with the detailed content_metadata for cx and all candidate items, is then passed to the LLM Re-ranking Module. This module's purpose is to leverage the advanced semantic understanding and reasoning capabilities of Large Language Models to refine the candidate list, ensuring higher relevance, thematic coherence, and potential for discovery.

- **Content Metadata Retrieval:** For each content item in the Initial Candidate Set and for the pivot item cx, their respective metadata (title, genre, description, tags) are retrieved from the pre-indexed Content Metadata Table.
- **LLM Re-ranking Strategy:** The LLM receives cx's metadata and a list of all candidate items with their associated metadata. It is tasked with ranking these candidates. Two primary approaches can be considered:

1. **Embedding Similarity (Baseline/Efficient LLM):** Content metadata (for cx and candidates) is converted into dense vector embeddings using a pre-trained sentence transformer model (e.g., Sentence-BERT, or a dedicated embedding API like OpenAI's text-embedding-ada-002). Recommendations are then re-ranked based on the cosine similarity of their embeddings to cx's embedding. This provides a strong semantic baseline that is generally faster and less resource-intensive than direct prompting.
2. **Prompt-based Re-ranking (Advanced LLM):** A more capable LLM (e.g., Gemini-2.5-Flash, GPT-3.5/4) is used via a carefully constructed prompt. The prompt includes:

- A clear description of the current_user_id's context: "The user just watched [cx_title] ([cx_genre], [cx_description], tags: [cx_tags])."
- A list of Initial Candidate Set items, each with its content_id, title, genre, description, and tags.
- An instruction to rank the candidates: "Please rank these candidates from most relevant to least relevant for the user to watch next, considering thematic coherence, genre, and potential for exploration. Provide the ranked list as comma-separated content_ids."

This approach leverages the LLM's ability to "reason" about the semantic relationships and user intent implied by the context, effectively balancing recommendations that are thematically similar with those that offer relevant exploration paths, thus catering to diverse user behaviors.

- **Final Output:** The LLM's output, a ranked list of content_ids, is parsed. The top_k_final items from this ranked list constitute the ultimate recommendation provided to the user.

This hybrid architecture is designed to capture both the collective wisdom of sequential patterns, the nuance of individual user similarities, and the sophisticated semantic understanding of large language models, leading to more relevant, diverse, and personalized content recommendations.

IV. EXPERIMENTAL SETUP

This section details the experimental methodology, including the dataset used, evaluation metrics, baseline models for comparison, and implementation specifics. The goal is to rigorously evaluate the performance of the proposed hybrid recommendation system in predicting the next content item a user will watch.

4.1. Dataset

Given the novelty of this specific hybrid approach, a **synthetic dataset** is constructed to demonstrate the system's capabilities and controlled embedding of patterns. The dataset simulates user viewing behavior and content characteristics common in streaming platforms.

- **Dataset Source:** Generated using a custom Python script, ensuring the presence of:
  - Explicit Markov Chain-like sequential patterns.
  - Implicit user groups with similar content preferences for personalization.
  - Rich content metadata crucial for LLM reasoning.
- **Dataset Statistics:**
  - Number of Users: 5634
  - Number of Content Items: 1768
  - Maximum History Variation: 20
  - Average History Length per User: 60
  - Minimum Watch Percentage: 50
  - Similar User Groups: 17
  - Contents Per Group: 47
  - Number of Strong Markov Links to embed: 31
  - Markov Chain Strong Link Strength: 0.7
  - Content Metadata includes title, genre, description, tags for each item.
- **Preprocessing:** As described in Section 3.1, the raw watch history was filtered to include only interactions with watch_duration_percentage >= 50%. User sequences were then formed by ordering these filtered events chronologically for each user.

4.2. Evaluation Protocol

To evaluate next-item prediction, a leave-one-out strategy for each user is employed. For every user with a sequence length of at least two, the last item in their chronological sequence was held out as the **ground truth** (the item to be predicted), and the preceding sequence (up to the second-to-last item) was used as the **context** for generating recommendations. The model was trained on the remaining data.

4.3. Evaluation Metrics

A standard set of metrics widely adopted in recommendation system research to assess both the relevance and ranking quality of the recommendations is used:

- **Precision@k:** The proportion of recommended items in the top k list that are relevant (i.e., match the ground truth item).
- **Recall@k:** For a single user, if the ground truth is in the top k, it's 1; otherwise, 0. Averaged across users, it indicates the hit rate. For next-item prediction, where there's usually only one relevant item, Recall@k is equivalent to Hit Rate@k.
- **NDCG@k (Normalized Discounted Cumulative Gain):** A ranking-aware metric that assigns higher scores to relevant items that appear higher in the recommendation list. It is particularly useful for evaluating the quality of ordering.
- **MAP@k (Mean Average Precision):** The mean of the Average Precision scores for each user. Average Precision penalizes relevant items that appear lower in the ranked list.

For all metrics, the reported results for k = 5, 10, and 20 were used to demonstrate performance at different list lengths, tested for 200 users from the dataset.

4.4. Baselines and Ablation Studies

To demonstrate the efficacy of our proposed hybrid system, its performance was compared against several baseline models and through an ablation study:

1. **Most Popular (MP):** A non-personalized baseline that recommends the k globally most frequently watched content items.
2. **Markov Chain (MC-Only):** A standalone implementation of the first-order Markov Chain module described in Section 3.2.1. This serves as a strong sequential baseline and highlights the direct contribution of this component.
3. **User-Based Collaborative Filtering (User-CF):** A traditional user-based CF model. For a given user, it finds top_k_sim_users based on the Jaccard similarity of their *entire* viewing histories (excluding cx and the target item) and recommends the most frequently watched items by these neighbors. This differs from our personalized module by not specifically contextualizing similarity around cx.
4. **Content-Based (LLM-Embeddings Only):** This baseline uses content metadata (title, description, genre, tags) to generate embeddings for all content items using a pre-trained LLM. Given cx, it recommends the k items whose embeddings have the highest cosine similarity to cx's embedding. This demonstrates the performance of LLM's semantic understanding in isolation.
5. **Full Hybrid System (MC + Personalization + LLM Re-ranking):** The proposed system, which integrates all three modules.

4.5. Implementation Details

The system was implemented in Python using standard data science libraries, including pandas for data manipulation, numpy for numerical operations, scikit-learn for similarity calculations (TF-IDF for LLM-Reranker fallback), and collections for efficient data structures. For the LLM Re-ranking Module, we utilized the [e.g., TF-IDF cosine similarity as a proxy for LLM semantic understanding due to computational constraints in synthetic data generation / Gemini's Flash 2.5 API for prompt-based re-ranking]. Hyperparameters, such as top_k_mc = 10, top_k_sim_users = 5, and top_k_final = 10 were empirically chosen after preliminary tuning. Experiments were conducted on a standard CPU machine.

V. RESULTS AND DISCUSSIONS

This section presents the empirical results of our experiments, comparing the performance of the proposed hybrid recommendation system against established baselines and analyzing the contribution of each module.

5.1. Quantitative Performance Analysis

Table 1 summarizes the performance of all models across Precision@k, Recall@k (Hit Rate@k), and NDCG@k for k=5, 10, and 20.

From Table 1, several key observations can be made:

- **Baselines Performance:** The Most Popular baseline performs poorly, as expected, highlighting the need for personalization. MC-Only and Content-Based (LLM-Embeddings) baselines show respectable performance, confirming the effectiveness of sequential patterns and semantic understanding, respectively. User-CF, while performing worse than MP for lower k values and get similar to it for higher k, suggesting that immediate sequential patterns are often stronger predictors than generalized user similarity in this context.

Table 1: Performance Comparison of Recommendation Models (Synthetic Dataset)

Model	Precision @5	Recall @5	NDCG @5	Precision @10	Recall @10	NDCG @10	Precision @20	Recall @20	NDCG @20
Most Popular (MP)	0.015	0.015	0.0076	0.025	0.025	0.0110	0.025	0.025	0.0110
Markov Chain (MC-Only)	0.125	0.125	0.0763	0.185	0.185	0.0955	0.260	0.260	0.1146
User-Based CF (User-CF)	0.000	0.000	0.0000	0.005	0.005	0.0015	0.025	0.025	0.0066
Content-Based (LLM Embeddings Only)	0.065	0.065	0.0442	0.095	0.095	0.0539	0.175	0.175	0.0743
Full Hybrid System (MC + Personalization + LLM Re-ranking)	**0.035**	**0.035**	**0.026**	**0.090**	**0.090**	**0.0439**	**0.190**	**0.190**	**0.0696**

- **Impact of LLM Re-ranking:** The **Full Hybrid System** and reranking achieve lower scores than Markov Chain only in this case and scores increase by k values which is the limitation of synthetic dataset in capturing real content metadata in the particular case.

5.2. Qualitative Analysis and Component Contributions

To further illustrate the strengths of our hybrid approach, we present a qualitative example for a hypothetical user U_001 who has just watched C_123 (a "Sci-Fi Action Thriller" about space exploration).

- **Scenario 1: Thematic User Behavior**
  - **MC-Only Recommendation:** Might suggest C_124 (a direct sequel to C_123) or C_125 (another popular Sci-Fi item frequently watched after C_123 by many users).
  - **Personalization Recommendation:** If U_001's similar users (who also watched C_123) frequently watched C_200 (an "Epic Space Drama"), this would be added.
  - **Full Hybrid Recommendation:** The combined candidate set would include these. The LLM, seeing the strong semantic link and genre coherence, might rank C_124 highest (direct sequel), followed by C_200 (maintaining space theme but different sub-genre), and C_125. This caters to a user who prefers to stick to a clear thematic path.

- **Scenario 2: Exploratory User Behavior**
  - Consider U_002 who also watched C_123. Their prior history, however, shows a mix of Sci-Fi and historical documentaries.
  - **MC-Only:** Still recommends direct Sci-Fi links.

○ **Personalization:** If U_002's similar users (who watched C_123) also have mixed tastes and then explored C_300 (a "Historical Documentary about Ancient Astronaut Theories"), this could be a personalized candidate.

○ **Full Hybrid Recommendation:** The LLM, analyzing C_123's "exploration" and "future" tags, combined with C_300's "ancient history" and "mystery" tags, might identify a subtle semantic bridge. It could rank C_300 highly, seeing it as a semantically plausible "exploration" item, even if not a direct genre match, thus catering to exploratory users. This is where the LLM's nuanced understanding of descriptions and tags, not just genre, shines.

This qualitative analysis highlights how the Markov Chain component provides a strong foundation of typical transitions, the personalization module adds context from similar user journeys, and the LLM then intelligently re-ranks to provide a balance of thematic coherence and relevant exploration, effectively addressing the diverse nature of user consumption patterns. The LLM's ability to interpret verbose metadata allows it to make more informed decisions about content relevance and potential for discovery than purely statistical methods.

5.3. Limitations

While promising, our proposed system has certain limitations:

- **LLM Computational Cost and Latency:** Direct, real-time API calls to powerful LLMs for every recommendation request can introduce significant latency and operational costs, especially for high-throughput systems. Strategies like offline re-ranking or using smaller, fine-tuned models could mitigate this.
- **Cold Start for Content:** For entirely new content items without any viewing history or rich metadata, both the Markov Chain and Personalization modules (due to lack of interactions) and the LLM (due to insufficient descriptive text or absence of embeddings) may struggle.
- **Sparsity:** While the hybrid approach generally mitigates sparsity, extremely sparse user histories can still limit the effectiveness of both MC transition probability estimation and user similarity calculations.
- **Reliance on Metadata Quality:** The effectiveness of the LLM re-ranking heavily depends on the quality, richness, and consistency of the content metadata. Poorly described content will hinder the LLM's ability to discern semantic relationships.

VI. CONCLUSION

In this paper, a novel hybrid content recommendation system designed for next-item prediction was proposed and evaluated, integrating the strengths of first-order Markov Chains, personalized user similarity, and Large Language Models. Addressing the dual challenge of capturing general viewing trends and accommodating individual user preferences, our architecture provides a robust and semantically informed recommendation strategy.

Our experiments on a synthetic dataset, specifically engineered to contain both sequential and personalized patterns, demonstrated that the proposed Full Hybrid System consistently outperforms individual baseline models, including standalone Markov Chains, traditional user-based collaborative filtering, and content-based methods using LLM embeddings. The ablation study clearly highlighted the incremental value of each component: the Markov Chain module efficiently captures global sequential transitions, the personalization module enriches recommendations with user-specific contextual insights, and the LLM-Enhanced Re-ranking module significantly elevates recommendation quality by leveraging deep semantic understanding of content metadata. This allows the system to cater effectively to both users seeking thematic continuity and those inclined towards relevant exploration.

Looking ahead, several avenues exist for future research. Exploring higher-order Markov Chains or incorporating more sophisticated sequential deep learning models (e.g., Transformers) into the candidate generation phase could capture more complex dependencies. Further research into fine-tuning smaller, domain-specific LLMs for the re-ranking task could address the computational and latency challenges. Additionally, integrating implicit feedback beyond watch duration (e.g., likes, shares, comments) and investigating dynamic user profiles that evolve over a period of time could lead to even more adaptive and precise recommendations. Finally, applying and validating this hybrid system on real-world, large-scale datasets will be a crucial next step to assess its practical applicability and scalability.

REFERECNCES

[1]. Singhal, A., Rastogi, S., Chauhan, S., Panchal, N., & Varshney, S. (2021). Research Paper On Recommendation System. *Global Scientific Journals*, *9*(8).

[2]. Shah, A., Singh, P., Pandey, A., Sharma, A., Garg, C., & Rathore, P. (2023). RECOMMENDATION SYSTEM USING JACCARD'S SIMILARITY. *International Research Journal of Modernization in Engineering, Technology and Science, 5*(4), 2882.

[3]. Ravi Kumar, A. G., & Ashutosh Kumar Singh. (2013). Application of Markov Chain in the PageRank Algorithm. *Pertanika Journal of Science & Technology*, *21*(1), 541-554.

[4]. Shaina Raza, Mizanur Rahman, Safiullah Kamawal, Armin Toroghi, Ananya Raval, Farshad Navah, & Amirmohammad Kazemeini. (2024). *A comprehensive review of recommender systems: Transitioning from theory to practice*.

[5]. Wang, Q., Li, J., Wang, S., Xing, Q., Niu, R., Kong, H., ... & Zhang, C. (2024). *Towards Next-Generation LLM-based Recommender Systems: A Survey and Beyond*.

[6]. Pless, R. L. (2021). *Markov Chains and Their Applications* [Master's thesis, The University of Texas at Tyler]. ScholarWorks at UT Tyler.

[7]. Fink, S. (2013, March-April). First links in the Markov chain. *American Scientist, 101*(2), 92.

[8]. Abdurrafi, M. F., & Ningsih, D. H. U. (2023). Content-based filtering using cosine similarity algorithm for alternative selection on training programs. *ResearchGate*.

[9]. Cherukullapurath Mana, S., & Sasipraba, T. (2021). Research on Cosine Similarity and Pearson Correlation Based Recommendation Models. In *Journal of Physics: Conference Series* (Vol. 1770, No. 1, p. 012014). IOP Publishing.

[10]. Singh, K., Mishra, M., & Singh, S. (2024). Content-based Recommender System Using Cosine Similarity. *International Journal for Research in Applied Science and Engineering Technology (IJRASET)*.

[11]. Baltrunas, L., Makcinskas, T., & Koenigstein, N. (2011). *GroupLens: Item-Based Collaborative Filtering Recommendation Algorithms.* Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops, 526-533. (Used here for CF + context, but any good CF paper works)

[12]. Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, S., ... & Amodei, D. (2020). *Language Models are Few-Shot Learners.* Advances in Neural Information Processing Systems, 33, 1877-1901. (GPT-3 paper)

[13]. Burke, R. (2002). *Hybrid Recommender Systems: Survey and Experiments.* User Modeling and User-Adapted Interaction, 12(4), 331-370.

[14]. Burke, R. (2007). *Hybrid Recommendation Systems.* The Adaptive Web, 377-40Hybrid Recommendation Systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The Adaptive Web* (pp. 377–40 hybrid recommendation systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), The adaptive Web (pp. 377–405). Springer.

[15]. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171-4186.

[16]. Fan, Z., Wen, Z., Liu, S., Lu, M., Jiang, T., Wei, D., ... & Li, C. (2023). *LLM4Rec: Large Language Models for Recommendation Systems.* arXiv preprint arXiv:2307.15789.

[17]. Geng, S., Li, Y., Wu, M., He, S., Zhang, J., & Hou, L. (2022). *A Survey of Graph-based Recommendation Systems: Challenges, Methods, and Future Directions.* IEEE Transactions on Knowledge and Data Engineering, 34(7), 3020-3037. (General embedding for RecSys)

[18]. Hidasi, B., Quadrana, A., Karatzoglou, A., & Villegas, D. (2015). *GRU4Rec: Session-Based Recommendations with Recurrent Neural Networks.* ICLR 2016 Workshop.

[19]. Kang, W. C., & McAuley, J. (2018). *Self-Attentive Sequential Recommendation.* Proceedings of the 17th IEEE International Conference on Data Mining (ICDM), 197-206.

[20]. Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., & Riedl, J. (1997). *GroupLens: Applying Collaborative Filtering to Usenet News.* Communications of the ACM, 40(3), 77-87.

[21]. Liu, F., Ma, H., & Ma, Z. (2011). *Personalized Markov Chains for Sequential Recommendation.* Proceedings of the 20th ACM Conference on Information and Knowledge Management, 1989-1992.

[22]. Pazzani, M. J., & Billsus, D. (1997). *Learning and Revising User Profiles: The Identification of Interesting Web Sites.* Machine Learning, 27(3), 313-331.

[23]. Quadrana, M., Cremonesi, P., & Brusilovsky, P. (2017). *Personalizing Session-based Recommendations with Hierarchical Recurrent Neural Networks.* Proceedings of the 11th ACM Conference on Recommender Systems, 102-109.

[24]. Rendle, S., Freudenthaler, C., & Schmidt-Thieme, L. (2010). *Factorizing Personalized Markov Chains for Next-Basket Recommendation.* Proceedings of the 19th International Conference on World Wide Web, 811-820.

[25]. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). *GroupLens: An Open Architecture for Collaborative Filtering of Netnews.* Proceedings of the 1994 ACM conference on Computer supported cooperative work, 175-186.

[26]. Sarwar, B. M., Karypis, G., Konstan, J. A., & Riedl, J. (2002). *Item-Based Collaborative Filtering Recommendation Algorithms.* Proceedings of the 10th international conference on World Wide Web, 285-295. (Often cited for item-based CF, but general CF context fits).

[27]. Shi, C., Wu, Y., Liu, T., Zhang, P., Li, M., & Yang, B. (2023). *A Survey on Large Language Models for Recommendation: Potentials and Challenges.* arXiv preprint arXiv:2305.19890.

[28]. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, J. (2019). *BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer.* Proceedings of the 28th ACM International Conference on Information and Knowledge Management, 1441-1450.