LlamaRec-LKG-RAG: A Single-Pass, Learnable Knowledge Graph-RAG Framework for LLM-Based Ranking

Vahid Azizi, Fatemeh Koochaki
{va.azizi, fatmakoochaki}@gmail.com

ABSTRACT

Recent advances in Large Language Models (LLMs) have driven their adoption in recommender systems through Retrieval-Augmented Generation (RAG) frameworks. However, existing RAG approaches predominantly rely on flat, similarity-based retrieval that fails to leverage the rich relational structure inherent in user-item interactions. We introduce LlamaRec-LKG-RAG, a novel single-pass, end-to-end trainable framework that integrates personalized knowledge graph context into LLM-based recommendation ranking. Our approach extends the LlamaRec architecture by incorporating a lightweight user preference module that dynamically identifies salient relation paths within a heterogeneous knowledge graph constructed from user behavior and item metadata. These personalized subgraphs are seamlessly integrated into prompts for a fine-tuned Llama-2 model, enabling efficient and interpretable recommendations through a unified inference step. Comprehensive experiments on ML-100K and Amazon Beauty datasets demonstrate consistent and significant improvements over LlamaRec across key ranking metrics (MRR, NDCG, Recall). LlamaRec-LKG-RAG demonstrates the critical value of structured reasoning in LLM-based recommendations and establishes a foundation for scalable, knowledge-aware personalization in next-generation recommender systems. Code is available at repository.

1 INTRODUCTION

Large Language Models (LLMs) have achieved state-of-the-art performance across diverse natural language processing tasks, including summarization, translation, and question answering Brown et al. (2020); Zhang et al. (2022); OpenAI (2023); Touvron et al. (2023). To enhance factual accuracy and domain adaptability, Retrieval-Augmented Generation (RAG) frameworks integrate LLMs with external retrieval mechanisms, enabling models to dynamically incorporate relevant contextual information during generation Lewis et al. (2020); Izacard & Grave (2021). Despite their effectiveness, traditional RAG approaches predominantly rely on flat retrieval strategies that treat documents or passages as isolated units, ranking them solely based on vector similarity measures. This approach fundamentally overlooks the rich semantic relationships and structural dependencies between information elements, leading to fragmented retrievals and potentially incomplete responses for complex, multi-faceted queries. Graph-based Retrieval-Augmented Generation (GraphRAG) emerges as a promising solution to these limitations by representing information as structured graphs, where nodes capture entities and edges encode explicit semantic or relational dependencies Asai & Hajishirzi (2021); Yasunaga et al. (2022); Wang et al. (2024a); Li et al. (2025); Sun et al. (2023a); Chen et al. (2024); Smith et al. (2025); Lee et al. (2025). This graph-structured representation enables sophisticated reasoning over interconnected information networks, facilitating multi-hop inference pathways and significantly improving the coherence and completeness of generated responses.

The rapid advancement of LLMs has positioned them as pivotal components in modern recommender systems, significantly enhancing personalization through robust representational capabilities, contextual reasoning, and generalization from limited data. The rapid advancement of LLMs has positioned them as pivotal components in modern recommender systems. Early methodologies primarily integrated LLMs as static modules, such as feature extractors or semantic enrichers, to refine user and item representations and process auxiliary information Cao et al. (2023); Xu et al.

(2023); Liu et al. (2024a); Wang et al. (2024b); Zhang et al. (2025). However, these approaches often underutilize the interactive and generative potentials of LLMs. Recent research has shifted towards treating LLMs as active agents within the recommendation pipeline, enabling roles such as retrievers, rankers, or fully generative recommenders capable of producing ranked lists and explanations Zhao et al. (2023a); Cinar et al. (2023); Zhang et al. (2024). This evolution facilitates advanced functionalities, including multi-turn dialogues, zero-shot personalization, and context-aware suggestions, steering recommender systems towards more conversational and explainable paradigms Liu et al. (2023); Gao et al. (2023); Zhang et al. (2023); Liu et al. (2024b); Chen et al. (2025); Wang et al. (2023). Despite significant progress, challenges such as cold-start, data sparsity, and intent ambiguity remain central in recommender systems. RAG methods, particularly GraphRAG, have emerged as practical solutions to address these issues. By retrieving structured meta-knowledge from user-item graphs, knowledge graphs (KGs), or behavioral traces and integrating it with LLMs, GraphRAG enables fine-grained reasoning under sparse or noisy conditions Yao et al. (2023); Sun et al. (2023b); Hou et al. (2023); Wang et al. (2025); Peng et al. (2024). GraphRAG has been applied to diverse recommendation tasks, including: Graph-based recommendation Sun et al. (2023b); Hou et al. (2023), Next-item prediction Zhao et al. (2023a); Cinar et al. (2023); Xu et al. (2023) and conversational recommendation Liu et al. (2023); Gao et al. (2023).

Integrating LLMs into recommendation systems introduces several challenges, including hallucinations, bias, limited controllability, lack of explainability, and high inference costs Zhao et al. (2023b); Lin et al. (2023); Ma et al. (2023); Said (2024). Recent research has explored scalable and interpretable solutions to mitigate these limitations. LlamaRec Yue et al. (2023a) improves scalability with a two-stage architecture: a lightweight retriever first selects candidate items, followed by a fine-tuned Llama-2-7b Touvron et al. (2023) model for ranking. It employs a *verbalizer*, a compact module that directly maps LLM logits to ranking scores in a single forward pass to reduce inference latency. To address hallucination and enhance reasoning in LLMs, Think-on-Graph (ToG) Sun et al. (2023a) introduces an agent-based framework where the LLM performs multi-hop inference by interactively traversing a KG Sun et al. (2023a). This dynamic process grounds predictions in explicit, factual paths, improving transparency and interpretability. While LlamaRec offers efficiency, it underutilizes the LLM's reasoning capabilities and relies on LLM's internal knowledge and data patterns. Conversely, ToG supports interpretable, structured reasoning but incurs latency due to iterative KG traversal, posing challenges for real-time, large-scale deployments for recommendation systems.

This work presents LlamaRec-LKG-RAG, an efficient and interpretable framework that integrates personalized information from KGs into LLM-based recommendation systems, aiming to enhance ranking performance. The proposed approach contributes two key innovations. First, we leverage the structured nature of recommendation data to construct an explicit KG comprising core entities, such as users, items, and ratings, alongside diverse relation types. This graph is further enriched with dataset-specific metadata, including item attributes (e.g., brand, category), to provide a richer semantic context. Second, we design a lightweight deep neural network called the user preference module to model sequential user-item interactions and capture dynamic user preferences over time. These preferences inform a relation-specific scoring function, enabling a precise, single-pass retrieval mechanism for extracting personalized subgraphs from the KG. The retrieved knowledge is then embedded into a carefully designed prompt template, combined with the user's historical interaction sequence and a candidate item set. This composite prompt is passed to a Llama-2-7b model, which performs the final ranking. The entire framework is trained end-to-end to optimize performance holistically, making LLM-based rankers integrate external personalized knowledge efficiently in a single pass and offering extensibility toward explainable recommendation systems. We evaluate our method against LlamaRec and report consistent improvements across standard recommendation benchmarks.

2   METHOD

We propose a novel and efficient methodology for extracting informative, user-personalized contextual information from KGs and seamlessly integrating this structured knowledge into LlamaRec's two-stage sequential recommendation framework to significantly enhance ranking performance. Our approach transforms the conventional LlamaRec architecture by incorporating a sophisticated knowledge graph reasoning component that enables personalized subgraph retrieval and integration.
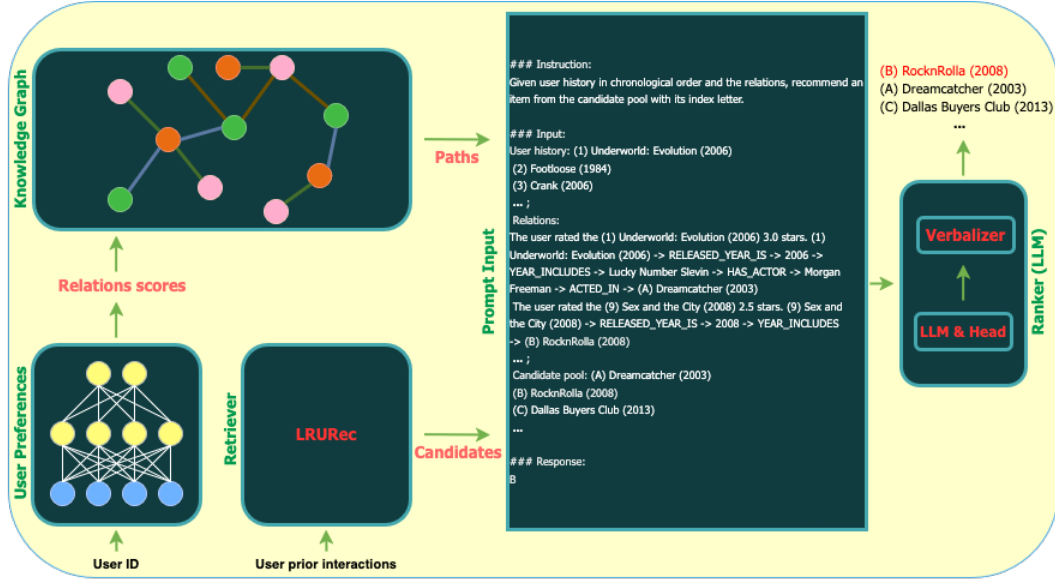
Figure 1: Overview of the LlamaRec-LKG-RAG framework. A sequential recommendation module (LRURec) generates candidate items from the user's interaction history. A user preference module captures dynamic interests to guide the retrieval of a personalized subgraph from a KG. The retrieved paths, user history, and candidate items are composed into a prompt, which is passed to a Llama-2-7b model for final ranking. The system is trained end-to-end to enable efficient and interpretable personalized ranking, augmented with user-specific knowledge extracted from the KG.

A comprehensive overview of the proposed LlamaRec-LKG-RAG architecture is presented in Fig. 1, with detailed methodological exposition provided in the following sections.

2.0.1 RETRIEVAL STAGE

The candidate generation stage processes a chronologically ordered sequence of user interactions to produce a refined set of candidate items for subsequent ranking. We employ LRURec Yue et al. (2023b;a), a computationally efficient sequential recommendation model built upon Linear Recurrent Units (LRUs) that maintains optimal performance while ensuring scalability. LRURec's architecture maintains a fixed-length hidden state representation, enabling efficient incremental updates and achieving low-latency inference capabilities essential for real-time recommendation deployment. The model is trained using an autoregressive objective that effectively captures complex user transition patterns and behavioral dynamics over time. During inference, LRURec generates refined item representations by computing dot products between predicted user state representations and learned item embeddings, enabling efficient similarity-based candidate identification. Given the extensive item catalog typical in recommendation scenarios, items are represented through unique identifier encodings to ensure computational tractability. We extract the top-K candidates ($K=20$) from LRURec's output distribution, providing a focused candidate set for the subsequent LLM-based ranking stage while maintaining computational efficiency.

2.0.2 RANKING STAGE

In the ranking stage following LlamaRec, we used the Llama-2-7b model. The LlamaRec authors created a prompt template Yue et al. (2023a) that includes task instructions, the user's recent interaction history, and candidate items as follows:

Instruction: Given user history in chronological order, recommend an item from the candidate pool with its index letter.
Input: User history: *history*; Candidate pool: *candidates*
Response: *label*

This format leverages item titles to help the model better understand user behavior and semantic signals. LlamaRec employs an innovative verbalizer-based ranking approach that assigns each candidate item a unique alphabetical index (A, B, C, etc.), transforming the ranking problem into a classification task. Rather than generating lengthy textual responses, the Llama-2-7b model efficiently identifies the most relevant item by outputting the corresponding index letter. The ranking scores are directly extracted from the output logits associated with these index tokens, enabling the entire ranking process to be completed in a single forward pass. This approach significantly enhances computational efficiency and system scalability while circumventing the computational overhead and potential ambiguities inherent in traditional listwise or pairwise ranking methodologies. The verbalizer-based design ensures deterministic, interpretable outputs while maintaining the semantic reasoning capabilities of the underlying language model.

2.0.3 LLAMAREC-LKG-RAG

Inspired by the ToG methodology, we have enhanced the LlamaRec framework by systematically integrating personalized contextual knowledge derived from structured knowledge graphs. This significantly improves the recommendation ranking performance. Our approach involves constructing a comprehensive, dataset-specific knowledge graph where users and items are the primary entities, connected through fundamental *RATED* relations that capture explicit user feedback and interaction patterns. To further enrich the knowledge graph semantically, we augment it with additional metadata, such as item attributes (e.g., brand). This results in a heterogeneous knowledge graph that incorporates multiple types of nodes and relations.

To enable efficient personalized traversal of this KG, we design a lightweight user preference model that learns user-specific preferences for different relation types. The user preference model identifies and prioritizes each user's most relevant relation types based on patterns learned through interactions over time. This model consists of: (1) An embedding layer to encode each user into a dense vector representation. (2) A fully connected layer to transform these embeddings. (3) A final fully connected layer with softmax activation to predict a probability distribution over relation types in the KG. We adopt the LlamaRec prompting structure to format input for LLM inference with the following template:

Instruction: Given user history in chronological order and the relations, recommend an item from the candidate pool with its index letter.
Input: User history: *history*; Relations: *Graph Paths*; Candidate pool: *candidates*
Response: *label*

Here, *history* and *candidates* are placeholders for user interaction history and candidate items, respectively. *Relations* correspond to personalized paths extracted from the KG based on user preferences. *label* represents the target item for each training example, which is left empty during inference for prediction. Our implementation focuses on retrieving semantic paths (information) between each history item and candidate item. Since multiple (or no) paths may exist, we extract the shortest path per pair with random tie-breaking to manage context size and remain within Llama-2's input limits. However, this still leads to up to $K \times K$ potential relation paths, where K is the number of items per set (see section 2.0.1), potentially exceeding input constraints. To address this, we use the output of the user preference model to score each relation type, thereby assigning a relevance score to each path. A naive approach would sum the relation scores along a path. However, this can introduce bias if certain high-frequency relations dominate. To mitigate this, we adopt a (Term Frequency-Inverse Document Frequency) TF-IDF-inspired weighting scheme: the model's learned relation scores are scaled by the TF-IDF score of each relation in the context of the current query. Each path's final score reflects personalization and informativeness, and only the top-K-scored paths are selected to be included in the LLM input context.

3 EXPERIMENTS

3.0.1 DATASET

We evaluated our approach using the ML-100K Harper & Konstan (2015) and Beauty He & McAuley (2016); McAuley et al. (2015) datasets. The ML-100K dataset consists of 100,000 user-item interactions and is a standard benchmark for movie recommendation systems. In contrast, the

Table 1: Summary statistics of the ML-100K and Beauty datasets, which include the number of users, items, interactions, and sequence lengths. Preprocessing follows the protocol in Yue et al. (2023a).

Datasets	Users	Items	Interactions	Length
ML-100K	610	3650	100K	147.99
Beauty	22332	12086	198K	8.87

Beauty dataset includes user reviews and interactions with beauty products collected from the Amazon platform. For preprocessing, we adhered to the protocol outlined in Yue et al. (2023a); Yang et al. (2023); Yue et al. (2022). This protocol involves constructing input sequences in chronological order, removing items that lack metadata, and filtering out users and items with fewer than five interactions. The statistics for both datasets are summarized in Table 1 Yue et al. (2023a).

3.0.2 KNOWLEDGE GRAPH (KG) CONSTRUCTION

This section outlines the construction process of a knowledge graph (KG) for each dataset. We employed Neo4j Neo4j (2024b) for KG creation and utilized Cypher Neo4j (2024a), Neo4j's declarative query language, to interact with the graph.

For the MovieLens dataset, the KG comprises four core entity types: users, movies (items), genres, and years. To enrich the graph with additional information, we extracted two more types of entities, directors and actors, from the IMDb website. The KG includes four primary types of relationships: (1) between users and movies, (2) between movies and actors, (3) between movies and directors, and (4) between movies and years. The user–movie relationship is represented as a unidirectional edge labeled *RATED*, forming triples of the form (User, *RATED*, Movie). In contrast, relationships involving movies and other entities (actors, directors, and years) are modeled bidirectionally to capture richer semantics. The relationship types are named to reflect their meaning, including:

- (Movie, *HAS_ACTOR*, Actor) and (Actor, *ACTED_IN*, Movie)
- (Movie, *DIRECTED_BY*, Director) and (Director, *IS_THE_DIRECTOR_OF*, Movie)
- (Movie, *RELEASED_YEAR_IS*, Year) and (Year, *YEAR_INCLUDES*, Movie)

The final MovieLens KG contains 10,471 nodes and 130,002 edges. An illustrative subgraph of the MovieLens KG structure is presented in Fig. 2.

The KG in the Beauty dataset comprises four primary entity types: users, items, product categories, and product brands. It includes five types of relationships among these entities. The user–item relationship is modeled as a unidirectional edge labeled *RATED*, forming triples such as (User, *RATED*, Item). Relationships between items, brands, and categories are modeled bidirectionally to capture semantic associations.

- (Item, *BRAND_IS*, Brand) and (Brand, *BRAND_INCLUDES*, Item)
- (Item, *CATEGORY_IS*, Category) and (Category, *CATEGORY_INCLUDES*, Item)

Additionally, the dataset contains a set of unidirectional item–item relationships that reflect user behavior patterns, including:

- (Item_X, *ALSO_BOUGHT*, Item_Y)
- (Item_X, *ALSO_VIEWED*, Item_Y)
- (Item_X, *BOUGHT_TOGETHER*, Item_Y)
- (Item_X, *BUY_AFTER_VIEWING*, Item_Y)

Each type of relation captures the intuitive semantics suggested by its name; for example, *ALSO_BOUGHT* indicates that two items were purchased frequently together. Due to the large size of the Beauty KG, querying during training and inference can be computationally intensive. To mitigate this, we applied two pruning strategies. First, because of its low frequency, we remove
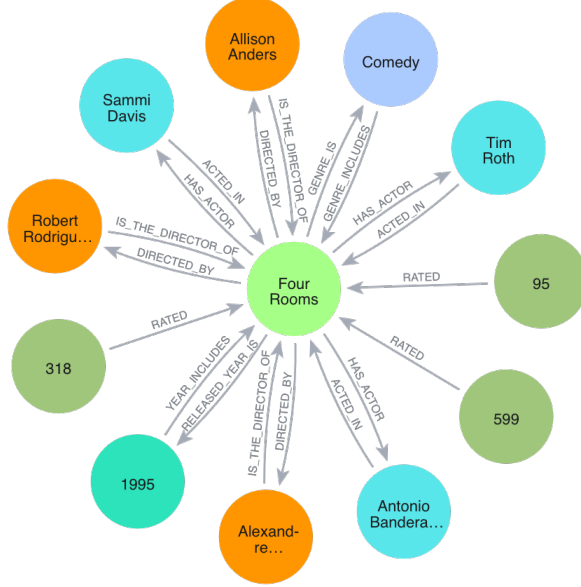
Figure 2: An illustrative subgraph of the knowledge graph constructed for the MovieLens dataset. The graph includes entity types such as users, movies, actors, directors, and years connected through semantically meaningful relationships. Unidirectional edges represent user ratings (e.g., (User, *RATED*, Movie)), while bidirectional edges capture contextual associations, such as acting, directing, and release year.
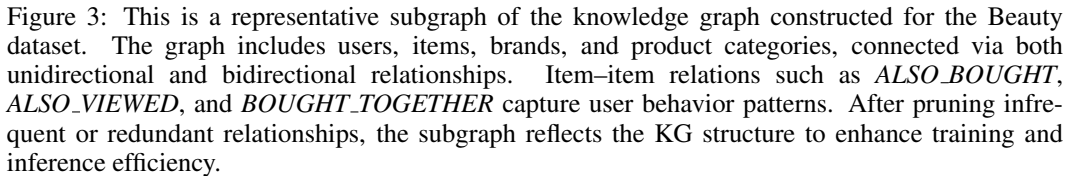
the *BUY_AFTER_VIEWING* relation. Second, for high-frequency relations such as *ALSO_VIEWED* and *BOUGHT_TOGETHER*, we retained only the most recent interaction per pair of items, preserving a single directional edge. Following pruning, the final KG contains 36,738 nodes and 543,088 relationships. A representative subgraph of Beauty KG is shown in Fig. 3.

3.0.3 BASELINE METHODS AND EVALUATION

In this study, we explore enhancing the ranking task with LLMs by efficiently integrating KGs. As our approach builds upon LlamaRec Yue et al. (2023a), we conduct a direct comparison exclusively against it. For evaluation, we adopt the leave-one-out strategy: for each user interaction sequence, the last item is reserved for testing, the second-to-last for validation, and the remaining items for training, following the protocol established in Yue et al. (2023a). To ensure a fair comparison, we employ the same evaluation metrics used in Yue et al. (2023a), including the mean reciprocal rank (MRR@K), the normalized discounted cumulative gain (NDCG@K) and the recall (Recall@K), where $K \in 1, 5, 10$.

3.0.4 IMPLEMENTATION

We employ LRURec Yue et al. (2023b;a) as the retriever module and train it separately for each dataset using the default hyperparameters provided in the original works Yue et al. (2023a;b). For model quantization, we adopt QLoRA Dettmers et al. (2023) applied to Llama-2, using the following LoRA Hu et al. (2021) configuration: rank dimension = 8, $\alpha = 32$, dropout = 0.05, and learning rate = 1×10^{-4}. The target modules are the Q and V projection matrices. Consistent with LlamaRec Yue et al. (2023a), we fix the number of items retained for the user history and candidate set at 20. Incorporating KG information increases the textual input length. Specifically, the input text length is set to 2,286 tokens for the MovieLens dataset and 2,536 tokens for the Beauty dataset. The maximum title length is capped at 32 tokens for MovieLens and 10 tokens for Beauty. The architecture of the user preference module is consistent across datasets. It consists of an embedding layer initialized using Xavier uniform initialization, followed by layer normalization Ba et al. (2016), a dropout layer (dropout rate = 0.2), a fully connected layer, and a softmax activation

Figure 3: This is a representative subgraph of the knowledge graph constructed for the Beauty dataset. The graph includes users, items, brands, and product categories, connected via both unidirectional and bidirectional relationships. Item–item relations such as *ALSO_BOUGHT*, *ALSO_VIEWED*, and *BOUGHT_TOGETHER* capture user behavior patterns. After pruning infrequent or redundant relationships, the subgraph reflects the KG structure to enhance training and inference efficiency.

for classification. For MovieLens, the user embedding dimension is set to 128, with four output classes corresponding to KG relation types: *GENRE_IS*, *RELEASED_YEAR_IS*, *DIRECTED_BY*, and *HAS_ACTOR*. For Beauty, the user embedding dimension is 512, with five output classes reflecting its KG relation types: *BRAND_IS*, *CATEGORY_IS*, *ALSO_BOUGHT*, *ALSO_VIEWED*, and *BOUGHT_TOGETHER*. All models are trained for one epoch and validated every 100 iterations. Early stopping is applied with a patience parameter of 20 iterations. The model checkpoint with the best validation performance is evaluated on the test set.

3.0.5 EVALUATION

The performance comparison between our model (LlamaRec-LKG-RAG) and LlamaRec is presented in Table 2, reporting results for MRR@K, NDCG@K, and Recall@K at $K \in 1, 5, 10$. Our model consistently outperforms LlamaRec across most metrics, although the degree of improvement varies between the two datasets. On the MovieLens dataset, our model demonstrates clear superiority across all evaluation metrics, with particularly notable performance gains. In contrast, the improvements in the Beauty dataset are more modest. At $K = 5$, for instance, both models yield nearly identical scores for NDCG and Recall. However, our model achieves approximately a 1% gain in MRR, which is especially meaningful since MRR emphasizes the rank of the first relevant item, aligning well with our ranking-centric objective. The relatively minor performance margin on the Beauty dataset may be attributed to its sparser interaction patterns, which limit the effectiveness of the user preference module. With fewer interactions per user, accurately modeling user behavior becomes more challenging, thereby reducing the impact of KG integration.

3.0.6 ABLATION STUDY

We conducted an additional experiment on the MovieLens dataset to evaluate the impact of incorporating KG context without utilizing the user preference module. In this variant, we extracted the shortest paths between each historical and candidate item pair, randomly selecting one in cases of multiple shortest paths, and included them as contextual input, denoted as *LlamaRec-KG-RAG*.

Figure 4: An illustrative example of a model query enriched with KG context selected using the user preference module. The correct answer *(B) RocknRolla (2008)* appears in multiple relation paths, particularly those involving the *RELEASED_YEAR_IS* relation. These consistent and semantically meaningful connections help the LLM reason effectively and make accurate predictions aligned with the user's inferred preferences.

Table 2: Performance comparison between LlamaRec-LKG-RAG and LlamaRec on the MovieLens and Beauty datasets. Results are reported for MRR@K, NDCG@K, and Recall@K at $K \in 1, 5, 10$. LlamaRec-LKG-RAG consistently outperforms the baseline, with notable gains on the MovieLens dataset, underscoring the effectiveness of incorporating KG information for the ranking task.

| | **ML-100K** | | **Beauty** | |
	LlamaRec-LKG-RAG	**LlamaRec**	**LlamaRec-LKG-RAG**	**LlamaRec**
MRR@1	**0.0262 ($\sim$22.5% ↑)**	0.0214	**0.0225 ($\sim$5% ↑)**	0.0214
NDCG@1	**0.0262 ($\sim$22.5% ↑)**	0.0214	**0.0225 ($\sim$5% ↑)**	0.0214
Recall@1	**0.0262 ($\sim$22.5% ↑)**	0.0214	**0.0225 ($\sim$5% ↑)**	0.0214
MRR@5	**0.0417 ($\sim$9% ↑)**	0.0383	**0.0349 ($\sim$1% ↑)**	0.0346
NDCG@5	**0.0499 ($\sim$7% ↑)**	0.0467	**0.0409 ($\sim$equal)**	0.0407
Recall@5	**0.0754($\sim$4.5% ↑)**	0.0721	**0.0593($\sim$equal)**	0.0594
MRR@10	**0.0462 ($\sim$7% ↑)**	0.0431	**0.0386 ($\sim$1.5% ↑)**	0.0380
NDCG@10	**0.0609 ($\sim$5% ↑)**	0.0580	**0.0498 ($\sim$1.5% ↑)**	0.0491
Recall@10	**0.1098 ($\sim$3% ↑)**	0.1065	**0.0868($\sim$1.5% ↑)**	0.0855

Figure 5: A prompt generated using KG context from *LlamaRec-KG-RAG* without the user preference module. The included relations between items are arbitrary and lack user-specific relevance, resulting in inconsistent and noisy contextual information. This unfiltered input hinders the model's reasoning ability and leads to incorrect predictions.

Table 3: Ablation study on the MovieLens dataset evaluating the impact of incorporating shortest paths between historical and candidate items as contextual input (*LlamaRec-KG-RAG*). The results show that introducing KG paths without a relevance filtering mechanism degrades performance, highlighting the critical role of the user preference module in selecting and integrating meaningful, user-personalized KG context. Bold values indicate the lowest performance scores across models.

	ML-100K		
	LlamaRec-LKG-RAG	LlamaRec-KG-RAG	LlamaRec
MRR@1	0.0262	**0.0180**	0.0214
NDCG@1	0.0262	**0.0180**	0.0214
Recall@1	0.0262	**0.1080**	0.0214
MRR@5	0.0417	**0.0324**	0.0383
NDCG@5	0.0499	**0.0382**	0.0467
Recall@5	0.0754	**0.0557**	0.0721
MRR@10	0.0462	**0.0380**	0.0431
NDCG@10	0.0609	**0.0517**	0.0580
Recall@10	0.1098	**0.9672**	0.1065

The results, shown in Table 3, indicate that injecting information without a filtering or relevance mechanism leads to a decline in model performance. This finding underscores the importance of integrating KG information selectively. In particular, it highlights the value of the user preference module, which captures user-specific behavior to identify the most relevant information. By personalizing the contextual input, the module ensures that only semantically meaningful and user-aligned information are incorporated, ultimately enhancing the effectiveness of the ranking model. Fig. 4 illustrates that the model prioritizes certain relation types for a given user, such as release year. In this example, the correct answer (Answer B) appears as an entity in multiple paths involving the year relation. Equipped with this structured context, the LLM can reason over these relations and infer that the user is most likely interested in the movie *RocknRolla (2008)*. In contrast, Fig. 5 presents the prompt for the same user, generated by *LlamaRec-KG-RAG*, which does not utilize the user preference module. As shown, the selected paths between items do not follow any coherent or user-aligned pattern. This lack of structure introduces noise and ambiguity, making it difficult for the model to perform practical reasoning. Consequently, the model is unable to identify the correct answer.

3.0.7 DISCUSSION

Several promising directions and configurations warrant further exploration. Below, we outline some of the most noteworthy avenues for future work.

This study utilized the Llama-2 model, which has a constrained context length of 4096 tokens and limited reasoning capacity. We conducted exploratory experiments with larger models, such as Chat-GPT OpenAI (2025) and Llama-2-70b Touvron et al. (2023), examining both configurations, with and without our proposed user preference module that filters paths versus using all $K \times K$ paths. Our preliminary results indicate that larger models perform well in both configurations. This suggests that models with longer input capacities and enhanced reasoning abilities may not require explicit information filtering. However, this approach poses two significant challenges: (1) extracting all $K \times K$ paths from the knowledge graph is computationally intensive and slow for real-time applications like recommendation systems, and (2) processing extended input sequences greatly increases inference costs.

One key benefit of our approach is its ability to generate explanations. We instructed LLMs, specifically ChatGPT, Llama-2-7b, and Llama-2-70b, to rank items and provide reasoning for their choices. Notably, all models, particularly the larger ones, produced coherent and plausible explanations. This experiment highlights a promising direction for future research on explainable recommendation systems leveraging LLMs and KGs Ai et al. (2018); Said (2024).

As mentioned, we utilized raw relation paths from the knowledge graph without verbalizing Han et al. (2025) them. However, converting KG triples and paths into natural language has been explored and presents an intriguing avenue for future work. This verbalization could enhance the

language model's understanding of the graph-based context, offering a rich area for further investigation.

We used a scoring strategy based on TF-IDF for path selection and ranking. Before implementing this approach, we tested various alternative heuristics, but they produced inconsistent results across our datasets. In contrast, TF-IDF consistently enhanced performance in both scenarios. Filtering strategies, such as excluding paths linked to low item ratings, can also affect performance. Conducting a systematic study of these filtering and ranking strategies is a promising avenue for future research.

In LlamaRec, item titles were utilized to guide ranking, drawing on the language model's internal knowledge. However, in specific datasets, this internal knowledge may be incomplete, making the use of item IDs a more suitable choice. In our experiments, our model (primarily relying on graph-based context) outperformed LlamaRec when using IDs instead of titles. This finding underscores the effectiveness of our approach when textual information is unreliable or unavailable, suggesting that further investigation in this area is warranted.

We augmented the KG with semantic metadata, such as brand information, but we did not include user-specific metadata in this study. Linking users to relevant nodes could incorporate user attributes, like age group in the MovieLens dataset, creating richer paths and potentially enhancing personalization. The effect of such user-level enrichment is an open question that warrants further research.

We extracted the paths between historical items and candidate items generated by LRURec, which outperformed traditional collaborative filtering methods. However, we did not examine how the quality of the retriever model influences overall performance. Since path extraction relies heavily on retrieved items, studying this interaction represents a valuable direction for future work.

Recent advancements in LLMs have significantly improved their reasoning capabilities, particularly through reinforcement learning-based fine-tuning techniques such as Reinforcement Learning from Human Feedback (RLHF). These techniques enable LLMs to perform multi-step reasoning, align with human intent, and generate contextually relevant and logically consistent outputs. Notably, these capabilities are starting to influence the design of next-generation recommendation systems. By integrating LLMs fine-tuned for reasoning, modern recommender frameworks can progress beyond static preference modeling to provide more interactive, context-aware, and goal-directed recommendations. Further integration and training of such frameworks using RLHF and Reinforcement Learning from AI Feedback (RLAIF) techniques could be a promising avenue, especially in live settings.

Finally, we constructed the KG offline before running our model. However, our framework is sequential, meaning temporal consistency must be maintained. Adding metadata, such as director information in the MovieLens dataset or brand details in the Beauty dataset, could inadvertently introduce paths containing future information, thereby violating causal constraints. To address this, a key direction for future work is to build the KG dynamically, filtering out nodes or edges with timestamps that exceed the current point in time. This approach would more accurately simulate real-world deployment and provide valuable insights into temporal generalization.

4 CONCLUSION

In this paper, we presented an efficient framework for integrating knowledge graphs (KGs) into LLM-based recommendation systems, with a focus on enhancing ranking performance. Our contributions are twofold: (1) we introduce the use of KGs as structured, contextual input to support LLM-based item ranking, and (2) we propose a lightweight neural network that models personalized user preferences to guide the extraction of relevant subgraphs from the KG. Building on LlamaRec, we augmented its two-stage recommendation pipeline with these components to enable efficient, interpretable, and personalized ranking. Extensive experiments on the MovieLens and Amazon Beauty datasets demonstrate that our approach consistently outperforms the baseline across standard recommendation metrics. These findings underscore the effectiveness of combining structured, user-aligned knowledge with LLMs and suggest promising directions for future work in scalable, knowledge-aware, and explainable recommendation systems.

REFERENCES

Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms*, 11(9), 2018. ISSN 1999-4893. doi: 10.3390/a11090137. URL `https://www.mdpi.com/1999-4893/11/9/137`.

Akari Asai and Hannaneh Hajishirzi. Edgeformer: A graph-based framework for multi-hop question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1414–1431, 2021.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL `https://arxiv.org/abs/1607.06450`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Yixin Cao, Xiangnan He, Yongfeng Zhang, et al. A survey of large language models for recommender systems. *arXiv preprint arXiv:2307.07069*, 2023.

Hao Chen, Wei Xu, and Ming Zhang. A survey on llm-powered agents for recommender systems. *arXiv preprint arXiv:2502.10050*, 2025. URL `https://arxiv.org/abs/2502.10050`.

L. Chen, W. Huang, and S. Liu. Think-on-graph 2.0: Deep and faithful large language model reasoning with knowledge-guided retrieval augmented generation. *arXiv preprint arXiv:2407.10805*, 2024. URL `https://arxiv.org/abs/2407.10805`.

Yasin Cinar, Pinar Karagoz, and Jundong Li. Llmrec: Personalized recommendation via prompting large language models. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3057–3061, 2023.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: efficient finetuning of quantized llms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Qifan Gao, Jiaqi Tang, and Minlie Huang. Chatgpt is all you need for conversational recommendation: Towards personalized preference elicitation with large language models. *arXiv preprint arXiv:2302.09104*, 2023.

Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A. Rossi, Subhabrata Mukherjee, Xianfeng Tang, Qi He, Zhigang Hua, Bo Long, Tong Zhao, Neil Shah, Amin Javari, Yinglong Xia, and Jiliang Tang. Retrieval-augmented generation with graphs (graphrag), 2025. URL `https://arxiv.org/abs/2501.00309`.

F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4), December 2015. ISSN 2160-6455. doi: 10.1145/2827872. URL `https://doi.org/10.1145/2827872`.

Ruining He and Julian McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pp. 507–517, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee. ISBN 9781450341431. doi: 10.1145/2872427.2883037. URL `https://doi.org/10.1145/2872427.2883037`.

Yifan Hou, Yajing Qi, Zhiwen Yu, Yanfang Song, Zhu Li, and Chengzhong Zhang. Towards reasoning-enhanced recommender systems: A survey and future directions. *arXiv preprint arXiv:2305.15706*, 2023.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL `https://arxiv.org/abs/2106.09685`.

Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*, 2021.

D. Lee, J. Park, and S. Kim. Gfm-rag: Graph foundation model for retrieval augmented generation. *arXiv preprint arXiv:2502.01113*, 2025. URL https://arxiv.org/abs/2502.01113.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kulkarni, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in neural information processing systems*, volume 33, pp. 9459–9474, 2020.

H. Li, Y. Chen, and M. Zhang. Grag: Graph retrieval-augmented generation for multi-hop reasoning. In *Findings of NAACL*, 2025. URL https://aclanthology.org/2025.findings-naacl.232.

Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. How can recommender systems benefit from large language models: A survey. *arXiv preprint arXiv:2306.05817*, 2023.

Jing Liu, Wei Chen, and Mei Huang. Llm-rec: Large language models for recommendation with interaction and reasoning. *arXiv preprint arXiv:2401.01234*, 2024a. URL https://arxiv.org/abs/2401.01234.

Yihong Liu, Lei Zheng, Yong Ge, Qi Li, and Enhong Zhang. Chat-rec: Towards interactive and explainable llms-augmented recommendation. *arXiv preprint arXiv:2305.14251*, 2023.

Zhen Liu, Lei Zhang, and Jing Gao. Large language model enhanced recommender systems: Taxonomy, trend, application and future. *arXiv preprint arXiv:2412.13432*, 2024b. URL https://arxiv.org/abs/2412.13432.

Tianhui Ma, Yuan Cheng, Hengshu Zhu, and Hui Xiong. Large language models are not stable recommender systems. *arXiv preprint arXiv:2312.15746*, 2023.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pp. 43–52, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450336215. doi: 10.1145/2766462.2767755. URL https://doi.org/10.1145/2766462.2767755.

Neo4j. *Cypher Query Language - Introduction*, 2024a. URL https://neo4j.com/docs/cypher-manual/current/introduction/. Accessed: 2025-05-12.

Neo4j. Neo4j - the world's leading graph database, 2024b. URL https://neo4j.com/. Accessed: 2025-05-12.

OpenAI. Gpt-4 technical report. Technical report, OpenAI, 2023. URL https://openai.com/research/gpt-4.

OpenAI. Chatgpt (may 2025 version). https://chat.openai.com, 2025. Large language model developed by OpenAI.

Boci Peng et al. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*, 2024. URL https://arxiv.org/abs/2408.08921.

Alan Said. On explaining recommendations with large language models: A review. *arXiv preprint arXiv:2411.19576*, 2024.

J. Smith, R. Kumar, and L. Zhang. Rgl: A graph-centric, modular framework for efficient retrieval-augmented generation on graphs. *arXiv preprint arXiv:2503.19314*, 2025. URL https://arxiv.org/abs/2503.19314.

Yujia Sun, Can Xu, Wayne Xin Zhao, Da Yin, Philip S. Yu, and Ji-Rong Wen. Think-on-graph: Structured reasoning with knowledge graphs for large language models. *arXiv preprint arXiv:2312.02948*, 2023a.

Ziqian Sun, Xinyu Wang, Wenqiang Zhang, and Dawei Yin. Graphrag: Retrieval-augmented generation meets knowledge graph for recommendation. *arXiv preprint arXiv:2310.06674*, 2023b.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

J. Wang, X. Li, and Y. Zhao. Gnn-rag: Graph neural retrieval for large language model reasoning. *arXiv preprint arXiv:2405.20139*, 2024a. URL https://arxiv.org/abs/2405.20139.

Rui Wang, Fan Li, and Ming Zhao. A survey on large language models for recommendation. *arXiv preprint arXiv:2305.19860*, 2023. URL https://arxiv.org/abs/2305.19860.

Shijie Wang, Wenqi Fan, Yue Feng, Xinyu Ma, Shuaiqiang Wang, and Dawei Yin. Knowledge graph retrieval-augmented generation for llm-based recommendation. *arXiv preprint arXiv:2501.02226*, 2025. URL https://arxiv.org/abs/2501.02226.

Yi Wang, Xin Zhao, and Qian Li. Chatrec: Towards conversational recommendation with large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 3456–3462, 2024b. URL https://dl.acm.org/doi/10.1145/XXXXXX.

Canran Xu, Bowen Du, Qinming Zhan, et al. Llmr: Large language model-augmented recommender system. *arXiv preprint arXiv:2306.05817*, 2023.

Fan Yang, Zheng Chen, Ziyan Jiang, Eunah Cho, Xiaojiang Huang, and Yanbin Lu. Palr: Personalization aware llms for recommendation, 2023. URL https://arxiv.org/abs/2305.07622.

Hanxiong Yao, Chuhan Wu, and Yongfeng Zhang. Retrieval-augmented generation for recommendation: A survey. *arXiv preprint arXiv:2305.01944*, 2023.

Michihiro Yasunaga, Xiang Ren, Percy Liang, and Jure Leskovec. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2022.

Zhenrui Yue, Huimin Zeng, Ziyi Kou, Lanyu Shang, and Dong Wang. Defending substitution-based profile pollution attacks on sequential recommenders. In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys '22, pp. 59–70, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392785. doi: 10.1145/3523227.3546770. URL https://doi.org/10.1145/3523227.3546770.

Zhenrui Yue, Sara Rabhi, Gabriel de Souza Pereira Moreira, Dong Wang, and Even Oldridge. Llamarec: Two-stage recommendation using large language models for ranking. *arXiv preprint arXiv:2311.02089*, 2023a.

Zhenrui Yue, Yueqi Wang, Zhankui He, Huimin Zeng, Julian McAuley, and Dong Wang. Linear recurrent units for sequential recommendation. *arXiv preprint arXiv:2310.02367*, 2023b.

Lei Zhang, Rui Sun, and Hao Tan. Generative recommendation: A survey of large language models in recommender systems. *ACM Computing Surveys*, 58(3):1–35, 2025. doi: 10.1145/XXXXXX.

Rui Zhang et al. Prompt-llmrec: Towards personalized recommendation via prompt tuning of large language models. *Proceedings of the Web Conference 2024*, 2024.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Sid Dewan, Murtaza Genc, Sunita Goel, Christina Guta, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Yongfeng Zhang et al. Llm agents for personalized recommendations: Reasoning, planning, and acting. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023.

Wayne Xin Zhao, Yiheng Chang, Jianing Yang, Jun Wang, and Ji-Rong Wen. Llm4rec: Revolutionizing recommendation with large language models. *arXiv preprint arXiv:2307.10649*, 2023a.

Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and Qing Li. Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046*, 2023b.