Large Language Model Empowered Logical Relations Mining for Personalized Recommendation

Yanchao Tan
College of Computer and Data Science,
Fuzhou University, Fuzhou, China
yctan@fzu.edu.cn

Wanzi Shao
College of Computer and Data Science,
Fuzhou University, Fuzhou, China
shaowz0302@163.com

Guofang Ma*
School of Computer Science and Technology,
Zhejiang Gongshang University, Hangzhou, China
maguofang@zjgsu.edu.cn

Carl Yang
Department of Computer Science,
Emory University, Atlanta, United States
j.carlyang@emory.edu

Abstract

In personalized recommendations, users often express complex logical requirements, involving the intersection of multiple preferences over heterogeneous graphs containing users, items, and external knowledge. Existing methods for mining logical relations face challenges in scalability and often overlook the semantics of relations, which are essential for uncovering higher-order connections and addressing incomplete relations within the graph. To tackle these challenges, we propose `RelRec`, a novel approach that leverages large language models (LLMs) to mine logical relations and satisfy users' logical requirements in personalized recommendation tasks. Specifically, the framework begins with the extraction of user-driven logical relations, followed by a rule-based logical relation mining module that integrates both semantic and structural information using the capabilities of LLMs. By uncovering higher-order logical relations, our approach effectively refines the heterogeneous graph for reasoning capacity and recommendation accuracy. Extensive experiments on real-world datasets demonstrate that `RelRec` significantly outperforms existing methods.

CCS Concepts

• **Information systems** → **Recommender systems**; **Language models**; **Query representation**.

Keywords

Personalized recommendation; Logical relations mining; Large language models; Rule mining

ACM Reference Format:
Yanchao Tan, Wanzi Shao, Guofang Ma, and Carl Yang. 2025. Large Language Model Empowered Logical Relations Mining for Personalized Recommendation. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3701716.3715574

*Guofang Ma is the corresponding author.

ACM ISBN 979-8-4007-1331-6/25/04
https://doi.org/10.1145/3701716.3715574

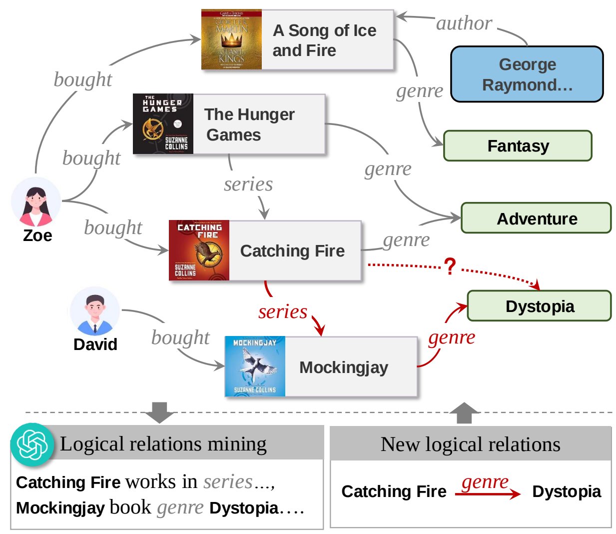

Figure 1: Illustration of the incomplete and implicit nature of relations in heterogeneous graphs.

1 Introduction

Personalized recommendation systems aim to deliver relevant suggestions tailored to users' preferences and needs [2, 5]. In many scenarios, users express complex logical requirements, such as the intersection of multiple preferences, which can be represented as structured queries over heterogeneous graphs comprising users, items, and external knowledge. These graphs capture both user-driven interactions and semantic relationships, providing a rich foundation for reasoning. However, effectively mining logical relations within such graphs to meet complex requirements remains a significant challenge [5, 11, 13].

Existing methods for logical relation mining in recommendation systems [8, 13] suffer from two key limitations. First, they often rely on computationally expensive graph traversal techniques to explore the rule space, which limits their scalability for large-scale graphs. Second, these methods frequently ignore the semantics of relations, which are crucial for uncovering higher-order logical connections

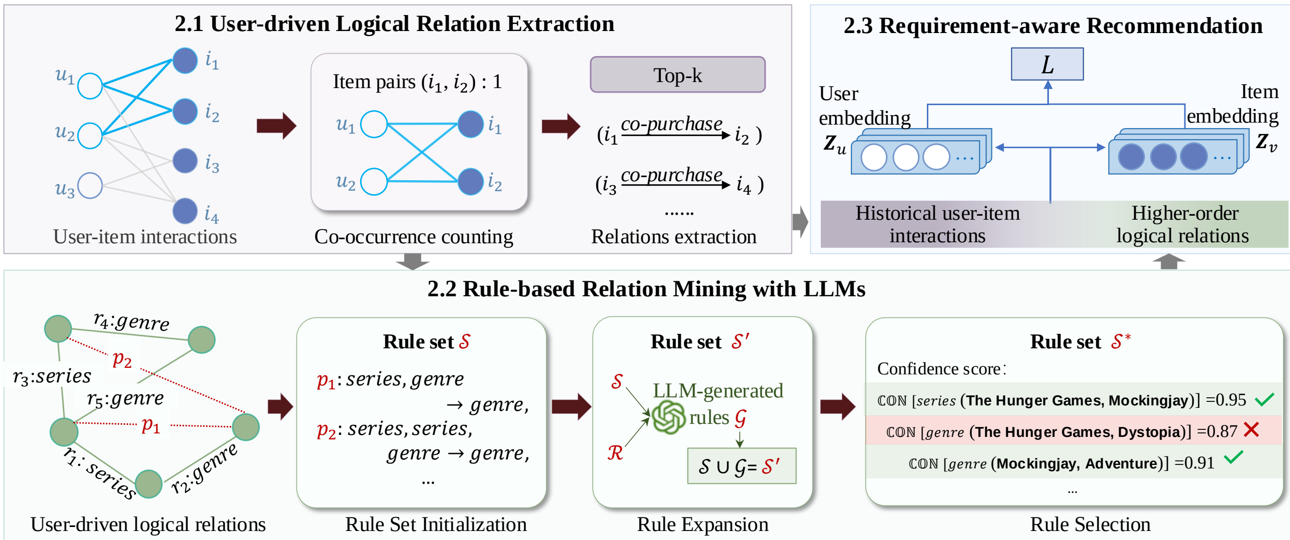

Figure 2: The overall framework of `RelRec`.

and addressing incomplete or implicit relations within the graph. For instance, as shown in Figure 1, **Mockingjay** from **The Hunger Games** series is primarily linked to the "*Dystopia*" genre due to its societal critique, while its "*Adventure*" aspects are ignored. This incomplete logical relation prevents the recommender system from fully satisfying user preferences, such as those of a user interested in both dystopian and adventure-themed content.

To tackle these challenges, we propose `RelRec` inspired by the recent success achieved by Large Language Models (LLMs) [5, 7, 12], a framework that leverages LLMs to mine logical relations efficiently. Specifically, `RelRec` begins with a user-driven logical relation extraction to identify implicit co-purchase associations. Then, we propose a rule-based relation mining with LLMs, which incorporates both semantic and structural information. By leveraging the powerful reasoning capabilities of LLMs, `RelRec` refines heterogeneous graphs, enabling the discovery of higher-order logical relations for accurate personalized recommendations. The overall framework of `RelRec` is illustrated in Figure 2.

Our main contributions can be summarized as follows:

- We propose a novel framework that effectively mines logical relations in heterogeneous graphs using LLMs, alleviating scalability and semantic problems in recommendation scenarios.
- We capture higher-order logical relations by integrating user-driven interactions with structural semantics, enhancing the graph's reasoning capacity for personalized recommendation.
- Extensive experiments demonstrate that `RelRec` significantly improves recommendation accuracy with an average of 17.36%, and can achieve scalable recommendation performance on unseen (zero-shot) logical requirements as well.

2 Related work

Recommendation systems have long relied on the classic method of collaborative filtering [1]. In recent years, methods that integrate background knowledge from knowledge graphs (KGs) have gradually attracted attention. These methods improve recommendation effects by enriching the representation of users and items [2, 17]. Different from existing methods, our research not only uses knowledge graphs as a source of background knowledge, but also further supports structured logical requirements provided by users, thereby injecting stronger interpretability and flexibility into the recommendation system.

Among the methods that use graphs to assist personalized recommendations, heterogeneous graph approaches use meta-paths to model multi-type relationships, but require manual path design and cannot handle incomplete [4]. Knowledge-enhanced methods [17] incorporated external knowledge yet treated relations as discrete labels without semantic grounding. In contrast, our framework enables more flexible and semantically rich reasoning on complex and incomplete knowledge graphs by leveraging LLMs for automatic rule mining.

3 The `RelRec` Framework

3.1 User-driven Logical Relation Extraction

KGs often suffer from sparse or incomplete item-item relations, limiting their ability to capture meaningful associations between items [5, 8]. To address this, we extract higher-order user-driven logical relations from the user-item graph by analyzing shared user interactions. For instance, if two items are frequently co-purchased by overlapping groups of users, these interactions indicate an implicit connection between the items—one that may not be directly given in the graph.

We model user-item interactions as a bipartite graph, where users $u \in \mathcal{U}$ and items $i \in \mathcal{I}$ are nodes, and interactions form edges $\mathcal{E}$. Higher-order relations are identified by closed loops that connect two users $u_1, u_2 \in \mathcal{U}$ and two items $i_1, i_2 \in \mathcal{I}$, satisfying:

$$i_1 \xleftrightarrow{\text{co-purchase}} i_2, \text{ s.t. } (u_1, i_1), (u_1, i_2), (u_2, i_1), (u_2, i_2) \in \mathcal{E}. \quad (1)$$

Here, the bidirectional relation reflects shared interactions between users u_1 and u_2, indicating meaningful co-purchase associations. We compute the frequency of all such item pairs (i_1, i_2) and select the top K pairs as the final user-driven logical relations.

3.2 Rule-based Relation Mining with LLMs

In personalized recommendations, capturing users' logical requirements within a heterogeneous graph is crucial. However, existing methods [8, 13] often overlook the semantics of relations. To address this, we propose a three-step process—rule initialization, expansion, and selection—leveraging LLMs to uncover higher-order relationships, thereby addressing incomplete relations and enabling more accurate recommendations.

Rule Set Initialization. The process begins with a breadth-first search to identify closed paths p in the graph. A path is defined as a sequence of relations $\{p : r_1, r_2, \ldots, r_n \rightarrow r\}$ that form a higher-order connection to emphasize how different relations are logically linked, where $r_i \in \mathcal{R}$ and $\mathcal{R}$ denotes the existing relation set from the heterogeneous graph. For example, a path could be $\{p_1 : series(r_1), genre(r_2) \rightarrow genre(r_2)\}$, where r_2 represents the target relation being inferred, and r_1, r_2 represent the supporting relations. The extracted rules are then used to form an initial rule set $\mathcal{S}$, which serves as the foundation for subsequent reasoning and knowledge augmentation.

Rule Expansion. While the initialized rule base $\mathcal{S}$ effectively captures explicit patterns, it remains constrained by the inherent incompleteness of the graph, which is especially critical in recommendation scenarios where relations among users and items are often sparse. To overcome this limitation, we leverage LLMs to expand the rule set by generating plausible logical rules that incorporate both structural and semantic information.

Specifically, given the initial rule set $\mathcal{S}$ and the existing relation set $\mathcal{R}$ from the original graph, we design a tailored prompting strategy to harness the reasoning capability of LLMs. Each rule from $\mathcal{S}$ is first verbalized into natural-language sentences by converting relation names into human-readable forms, which enhances the LLM's semantic understanding. For example, relations like "inv_bought" are verbalized as "inverse of bought." These verbalized rules, along with user-driven logical relation (cf. Section 3.1), are placed into a carefully crafted prompt template and fed into the LLM to generate additional candidate rules. The detailed prompt template for rule mining is provided in Section 4.1.

The final expanded rule set $\mathcal{S}'$ is formed by combining the initial rules $\mathcal{S}$ with the LLM-generated rules $\mathcal{G}$ as:

$$\mathcal{S}' = \mathcal{S} \cup \mathcal{G}. \tag{2}$$

By incorporating LLM-generated rules, our approach efficiently uncovers higher-order semantic patterns that are otherwise missed by traditional graph-based methods. This process eliminates the need for exhaustive graph searches and plays a significant role in recommendation tasks, where sparse user-item interactions can benefit significantly from enriched logical relations.

Rule Selection. To ensure the reliability of the rule set $\mathcal{S}'$, we further propose to evaluate their quality by designing a confidence score $\mathbb{CON}[r(\mathbf{x}, \mathbf{y})]$, where $r(\mathbf{x}, \mathbf{y})$ denotes entity $\mathbf{x}$ is connected to entity $\mathbf{y}$ through the relation r (e.g., the rule *series*(**The Hunger Games**, **Catching Fire**) in Figure 1). A rule $r_1(\mathbf{x}, \mathbf{z}_1), r_2(\mathbf{z}_1, \mathbf{z}_2), \ldots, r_n(\mathbf{z}_{n-1}, \mathbf{y}) \rightarrow r(\mathbf{x}, \mathbf{y})$ is validated based on:

$$\mathbb{CON}[r(\mathbf{x}, \mathbf{y})] = \frac{\#\{(\mathbf{x}, \mathbf{y}) \text{ s.t. } r_1(\mathbf{x}, \mathbf{z}_1), \ldots, r_n(\mathbf{z}_{n-1}, \mathbf{y}) \wedge r(\mathbf{x}, \mathbf{y}) \in \mathcal{S}'\}}{\#\{(\mathbf{x}, \mathbf{y}) \text{ s.t. } r_1(\mathbf{x}, \mathbf{z}_1), \ldots, r_n(\mathbf{z}_{n-1}, \mathbf{y})\}}. \tag{3}$$

Here, the numerator represents the number of relations where $r_1, r_2, \ldots, r_n$ are satisfied alongside $r(\mathbf{x}, \mathbf{y})$ within the expanded rule set $\mathcal{S}'$. The denominator represents all possible instances where $r_1, r_2, \ldots, r_n$ are satisfied. Rules with $\mathbb{CON}$ scores exceeding 0.9 are selected to form the final relation set $\mathcal{S}^*$, and the corresponding logical relations are incorporated into the graph. This enriched graph combines user-item interactions with higher-order logical connections, creating a more complete structure for personalized recommendation tasks.

Table 1: Statistics of the datasets used in our experiments.

	Amazon-Book	Last-FM	MIND
Users	70,679	23,566	100,000
Items	24,915	45,123	30,577
Interactions	847,733	3,034,796	2,975,319
Entities	88,572	58,266	24,733
Relations	39	9	512
Triplets	2,557,746	464,567	148,568

3.3 Requirement-aware Recommendation

To incorporate the refined rule set $\mathcal{S}^*$ into the recommendation process, users and items are represented by their embeddings $Z_u \in \mathbb{R}^d$ and $Z_v \in \mathbb{R}^d$ in a shared latent space. These embeddings are jointly learned by combining user preferences, derived from historical user-item interactions, and higher-order logical relations captured in the refined rule set $\mathcal{S}^*$, enabling the seamless integration of user-driven requirements and mined logical connections.

Then, the similarity between a user u and an item v is defined as:

$$p_{u,v} = \sigma\left(Z_u^\top Z_v + f(F_{u,v}; \mathcal{S}^*)\right), \tag{4}$$

where $Z_u^\top Z_v$ measures the direct similarity between the user and item embeddings, and $f(F_{u,v}; \mathcal{S}^*)$ incorporates logical relations derived from the refined rule set $\mathcal{S}^*$. f is a single-layer feed-forward network with softmax activation. $\sigma(\cdot)$ denotes the sigmoid function, ensuring the output is a probability score.

To train the model, we minimize the binary cross-entropy loss over positive and negative user-item pairs:

$$L = -\frac{1}{|\mathcal{D}|} \sum_{(u,v) \in \mathcal{D}} \left[y_{uv} \log(p_{uv}) + (1 - y_{uv}) \log(1 - p_{uv})\right], \tag{5}$$

where $y_{uv} \in \{0, 1\}$ indicates whether user u interacted with item v, and $\mathcal{D}$ represents the set of all training pairs.

4 Experiment

4.1 Experimental Setup

Datasets. In this experiment, we selected three diverse and widely used real-world datasets to evaluate the performance of our proposed method: Amazon-Book, Last-FM, and MIND. We summarize the statistics of three datasets in Table 1. These datasets cover a range of domains, including e-commerce, music streaming, and

Table 2: Recommendation performance on Amazon-book, Last-FM, and MIND. The best performances are highlighted in boldface. The second-best performances are underlined.

Method	Amazon-Book				Last-FM				MIND			
	H@10	H@20	N@10	N@20	H@10	H@20	N@10	N@20	H@10	H@20	N@10	N@20
RippleNet	0.070	0.109	0.031	0.040	0.390	0.482	0.201	0.233	0.092	0.123	0.054	0.061
CFKG	0.095	0.145	0.043	0.053	0.527	0.601	0.288	0.307	0.125	<u>0.161</u>	0.074	0.082
KGCN	0.048	0.079	0.019	0.025	0.266	0.346	0.124	0.148	0.062	0.085	0.031	0.035
MKR	0.040	0.067	0.016	0.022	0.223	0.296	0.104	0.125	0.050	0.071	0.027	0.033
GQE	0.084	0.114	0.052	0.058	0.468	0.502	0.340	0.337	0.111	0.127	0.089	0.090
LogicRec	<u>0.100</u>	<u>0.141</u>	<u>0.059</u>	<u>0.070</u>	<u>0.558</u>	<u>0.622</u>	<u>0.389</u>	<u>0.405</u>	<u>0.131</u>	0.156	<u>0.099</u>	<u>0.106</u>
`RelRec`	**0.136**	**0.177**	**0.082**	**0.090**	**0.562**	**0.624**	**0.398**	**0.413**	**0.162**	**0.206**	**0.107**	**0.118**

online news recommendation. Each dataset offers a unique context with distinct characteristics, providing a comprehensive and robust evaluation of `RelRec` across various recommendation scenarios.

Metrics. We used two metrics for evaluation: H@k (Hit Ratio at k) measures the accuracy of the recommendations by evaluating whether relevant items appear in the Top-k recommended list. N@k (Normalized Discounted Cumulative Gain at k), considers both the ranking and relevance of the recommended items.

Baselines. We adopt the following representative state-of-the-art baselines for comparison, which include two types. (1) **Knowledge Graph-based Methods:** These methods enhance the representation of users and items by integrating external knowledge from knowledge graphs. By leveraging rich relationships and information in the graph, these methods aim to improve recommendation accuracy and personalization. Notable methods include:

- **RippleNet**[14]: Propagates user preferences through the knowledge graph to improve recommendation relevance.
- **CFKG**[2]: Jointly learns recommendation and knowledge graph completion to better understand user preferences.
- **KGCN**[16]: Utilizes Knowledge Graph Convolutional Networks (KGCN) to capture inter-item relationships and address the sparsity issue in recommendation systems.
- **MKR**[15]: Proposes a multi-task learning framework that integrates knowledge graph embeddings for improved recommendation quality.

(2) **Logic Query Expansion Methods:** These methods focus on refining recommendations by expanding logical queries based on structured user inputs. They aim to improve recommendation systems by interpreting and processing complex logical queries. Key methods include:

- **GQE**[6]: Uses low-dimensional embeddings to predict conjunctive logical queries on incomplete knowledge graphs.
- **Q2B**[9]: Reasons over arbitrary logical queries in large, incomplete knowledge graphs.
- **BetaE**[10]: Answers arbitrary first-order logic queries over knowledge graphs while modeling uncertainty.
- **FuzzQE**[3]: A fuzzy logic-based query embedding framework for answering first-order logic queries without requiring complex training data.
- **LogicRec** [13]: Addresses users' complex logical requirements by using logical query embedding and a multi-task knowledge sharing mechanism.

Implementation Details. In our experiments, we intentionally leave out 5% of KG to simulate the presence of missing facts. At the same time, we carefully designed a prompt template to enable LLM to perform rule mining accurately, as follows:

> Simplified Prompt Template for Rule Mining
>
> Logical rules define relationships between two entities **X** and **Y** as logical implications, where the right-hand side is inferred from the left-hand side.
> Rule Samples: {series, genre → genre} . . .
> Generate the most important rules for $relation_i(\mathbf{X}, \mathbf{Y})$. Return rules only, no explanations.

4.2 Results with Personalized Recommendation

The experimental results are summarized in Table 2. `RelRec` consistently outperforms state-of-the-art baseline methods across all datasets and evaluation metrics, demonstrating its superiority in addressing personalized recommendation tasks. Specifically, `RelRec` surpasses knowledge graph-based baselines (e.g., CFKG) by incorporating logical queries to model users' requirements, enabling more precise intent modeling. Additionally, `RelRec` outperforms logic query expansion methods (e.g., LogicRec) by using LLMs to mine logical relations in heterogeneous graphs, addressing scalability and semantic limitations. Moreover, our proposed `RelRec`'s performance gains range from substantial (32.99% on Amazon-Book dataset) to modest (1.33% on Last-FM dataset). This variability underscores two key factors. Firstly, in highly sparse datasets, `RelRec` effectively overcomes the challenges posed by knowledge graph incompleteness. Secondly, in datasets with simpler relational structures, the limited number of logical rules generated by the LLMs may constrain the model's potential. Despite this limitation, `RelRec` consistently outperforms the second-best model, demonstrating its robustness across datasets with diverse characteristics.

4.3 Results with Logical Requirements

To demonstrate the effectiveness of our model over complex logical requirements, we evaluate `RelRec` over varying levels of logical complexity, which is consistent with [13]. As shown in Figure 3, the four key logical requirements simulate diverse user preferences and scenarios in recommendation tasks, where the symbols i, u, and p represent intersection (∧), union (∨), and projection, respectively.

Table 3: Experimental results of both basic and zero-shot logical requirements. We use H@20 as evaluation metric.

Methods	Basic Logical Req				Zero-shot Req			
	1p	2p	2i	3i	pi	ip	2u	up
	Amazon-Book							
GQE	0.071	0.055	0.170	0.256	0.165	0.070	0.081	0.079
Q2B	0.033	0.042	0.157	0.291	0.156	0.054	0.058	0.070
BetaE	0.033	0.050	0.154	0.301	0.158	0.033	0.065	0.053
FuzzQE	0.088	0.066	0.194	0.273	0.155	0.084	0.077	0.073
LogicRec	0.092	0.063	0.215	0.351	0.187	0.071	0.116	0.081
RelRec	**0.160**	**0.100**	**0.301**	**0.405**	**0.198**	**0.106**	**0.158**	**0.087**
	MIND							
GQE	0.008	0.025	0.265	0.271	0.373	0.029	0.097	0.035
Q2B	0.005	0.019	0.245	0.311	0.351	0.021	0.073	0.031
BetaE	0.006	0.024	0.243	0.323	0.356	0.015	0.077	0.023
FuzzQE	0.010	0.026	0.303	0.291	0.350	0.031	0.093	0.034
LogicRec	0.011	0.027	0.335	0.373	**0.422**	0.028	0.139	0.036
RelRec	**0.071**	**0.066**	**0.409**	**0.455**	0.417	**0.085**	**0.197**	**0.084**

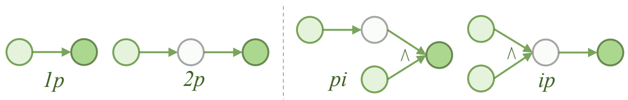

Figure 3: Examples of the basic logical requirements (left) and zero-shot logical requirements (right).

The experimental results, as shown in Table 3, indicate that the proposed RelRec method consistently outperforms the second-runner on Amazon-Book, with the improvement gains ranging from 15.38% to 73.91%.

On the MIND dataset, LogicRec achieves competitive results and slightly outperforms RelRec in the *pi* query on the MIND dataset, likely due to its multi-task knowledge-sharing mechanism, which effectively leverages both requirement-item and preference-item pairs for simpler intersection-based queries. However, its reliance on logical query embedding limits its ability to capture higher-order relations in incomplete graphs. FuzzQE also performs well in the *ip* query, leveraging its fuzzy logic mechanism to capture approximate relations and soft reasoning, making it effective for projection-heavy queries. However, its lack of integration for complex semantic patterns restricts its performance in more intricate scenarios. Overall, RelRec excels in capturing higher-order logical relations and refining heterogeneous graphs through LLM-based rule mining, enabling it to outperform LogicRec and FuzzQE in most scenarios. This highlights RelRec's ability to effectively address complex and zero-shot logical requirements for recommendations.

5 Conclusion

In this paper, we presented RelRec, a framework leveraging large language models to mine logical relations for personalized recommendations. By extracting user-driven relations and employing rule-based logical mining with LLMs, RelRec uncovers higher-order connections in heterogeneous graphs. In this way, we can integrate semantics with structural information for personalized recommendations. Experimental results show that RelRec achieves improved recommendation accuracy along with satisfying complex logical requirements.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants (No.62302098), Zhejiang Provincial Natural Science Foundation of China (LQ23F020007), Zhejiang Provincial Department of Agriculture and Rural Affairs Project (2024SNJF044), and Zhejiang Gongshang University "Digital+" Disciplinary Construction Management Project (SZJ2022B001). This research project has benefitted from the Microsoft Accelerating Foundation Models Research (AFMR) grant program and the OpenAI Researcher Access program. Carl Yang was not supported by any funds from China.

References

[1] T. F. Boka, Z. Niu, and R. B. Neupane. A survey of sequential recommendation systems: Techniques, evaluation, and future directions. *Information Systems*, 125:102427, 2024.

[2] Y. Cao, X. Wang, X. He, Z. Hu, and T. Chua. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The World Wide Web Conference*, pages 151–161, 2019.

[3] X. Chen, Z. Hu, and Y. Sun. Fuzzy logic based logical query answering on knowledge graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3939–3948, 2022.

[4] X. Fu, J. Zhang, Z. Meng, and I. King. Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding. In *The World Wide Web Conference*, pages 2331–2341, 2020.

[5] S. Geng, Z. Fu, J. Tan, Y. Ge, G. de Melo, and Y. Zhang. Path language modeling over knowledge graphsfor explainable recommendation. In *The World Wide Web Conference*, pages 946–955, 2022.

[6] W. L. Hamilton, P. Bajaj, M. Zitnik, D. Jurafsky, and J. Leskovec. Embedding logical queries on knowledge graphs. In *NeurIPS 2018*, volume 31, pages 2030–2041, 2018.

[7] L. Luo, J. Ju, B. Xiong, Y.-F. Li, G. Haffari, and S. Pan. Chatrule: Mining logical rules with large language models for knowledge graph reasoning. *arXiv preprint arXiv:2309.01538*, 2023.

[8] W. Ma, M. Zhang, Y. Cao, W. Jin, C. Wang, Y. Liu, S. Ma, and X. Ren. Jointly learning explainable rules for recommendation with knowledge graph. In *The World Wide Web Conference*, pages 1210–1221, 2019.

[9] H. Ren, W. Hu, and J. Leskovec. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In *8th International Conference on Learning Representations*, 2020.

[10] H. Ren and J. Leskovec. Beta embeddings for multi-hop logical reasoning in knowledge graphs. volume 33, pages 19716–19726, 2020.

[11] Y. Tan, H. Lv, Z. Zhou, W. Guo, B. Xiong, W. Liu, C. Chen, S. Wang, and C. Yang. Logical relation modeling and mining in hyperbolic space for recommendation. In *2024 IEEE 40th International Conference on Data Engineering*, pages 1310–1323. IEEE, 2024.

[12] Y. Tan, Z. Zhou, H. Lv, W. Liu, and C. Yang. Walklm: A uniform language model fine-tuning framework for attributed graph embedding. In *NeurIPS 2023*, 2023.

[13] Z. Tang, G. Floto, A. Toroghi, S. Pei, X. Zhang, and S. Sanner. Logicrec: Recommendation with users' logical requirements. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2129–2133, 2023.

[14] H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, and M. Guo. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 417–426, 2018.

[15] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, and M. Guo. Multi-task feature learning for knowledge graph enhanced recommendation. In *The World Wide Web Conference*, pages 2000–2010, 2019.

[16] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo. Knowledge graph convolutional networks for recommender systems. In *The World Wide Web Conference*, pages 3307–3313, 2019.

[17] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958, 2019.