Knowledge is Power: Boosting Recommender Systems by Infusing LLMs with Domain Expertise

Alessandro Petruzzelli[1], Cataldo Musto[1], Marco de Gemmis[1], Pasquale Lops[1] and Giovanni Semeraro[1]

Abstract

Large Language Models (LLMs) have emerged as a powerful new paradigm for recommender systems. However, their effectiveness is often constrained by the general-purpose knowledge acquired during pre-training, which may lack the domain-specific detail required for specialized recommendation tasks. To address this, we introduce a comprehensive pipeline for injecting multi-source knowledge directly into an LLM. Our methodology extracts and lexicalizes information from item descriptions (textual), knowledge graphs (structured), and user-item interactions (collaborative). This external knowledge is then infused into the model through a unified fine-tuning process that simultaneously adapts the LLM to a top-k re-ranking task. We conduct extensive experiments across movie, music, and book domains, demonstrating that our approach significantly enhances recommendation accuracy, especially in domains less-covered by the LLM's original training data. Our knowledge-injected model achieves state-of-the-art performance, outperforming a wide array of baselines, including powerful zero-shot models like GPT-4, in the music and book domains. This paper serves as a discussion of the research originally presented in the paper referenced as [12].

Keywords

Recommender Systems, Large Language Models, Knowledge Injection, Fine-Tuning, Domain Adaptation, Knowledge-Aware Systems

1. Introduction

The evolution of recommender systems has progressed from collaborative filtering [7], which suffers from data sparsity, towards Knowledge-Aware Recommender Systems (KARS) that leverage external data [4]. The latest paradigm shift involves Large Language Models (LLMs), which offer unprecedented zero-shot reasoning capabilities [11].

Current LLM-based recommendation strategies fall into two categories. **Non-tuning** approaches use pre-trained models like GPT-4 as-is, relying on sophisticated prompt engineering to elicit recommendations [15]. This method is limited by the LLM's static, general-purpose knowledge. **Tuning** approaches adapt smaller, open-source LLMs (e.g., LLaMA [14]) to recommendation tasks via instruction tuning [5]. While frameworks like P5 [5] unify various recommendation tasks into a text-to-text format, they primarily focus on task adaptation rather than enriching the model's core knowledge base.

We identify a critical gap: the need to explicitly infuse LLMs with curated, domain-specific knowledge. This process, which we term *knowledge injection*, is vital for enhancing the model's understanding of items, particularly in niche domains (e.g., technical books, indie music) that are underrepresented in general pre-training corpora.

This paper introduces a novel pipeline for injecting multi-source knowledge into an LLM for top-k recommendation. Our contributions are:

1. A modular pipeline for extracting, lexicalizing, and injecting knowledge from textual descriptions, knowledge graphs, and collaborative signals into an LLM.

IIR2025: 15th Italian Information Retrieval Workshop, 3th - 5th September 2025, Cagliari, Italy

*Corresponding author.

alessandro.petruzzelli@uniba.it (A. Petruzzelli); cataldo.musto@uniba.it (C. Musto); marco.degemmis@uniba.it (M. d. Gemmis); pasquale.lops@uniba.it (P. Lops); giovanni.semeraro@uniba.it (G. Semeraro)

0009-0008-2880-6715 (A. Petruzzelli); 0000-0001-6089-928X (C. Musto); 0000-0002-2007-9559 (M. d. Gemmis); 0000-0002-6866-9451 (P. Lops); 0000-0001-6883-1853 (G. Semeraro)

2. A comprehensive analysis across three domains demonstrating how different knowledge types impact recommendation accuracy.
3. Evidence that our knowledge-injected model achieves state-of-the-art performance, outperforming strong baselines, including GPT-4, particularly in specialized domains.

2. Methodology

Our goal is to improve top-k item recommendation by training an LLM to re-rank a candidate list of items for a user u. The methodology involves two primary phases: a unified training stage for knowledge injection and task adaptation, followed by an inference stage. A detailed rappresentation of our methodology is illustrated in Figure 1

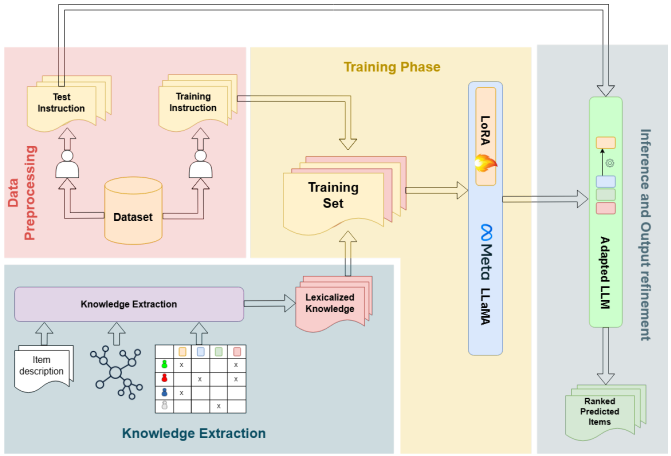

Figure 1: High-level architecture of the proposed knowledge injection and recommendation pipeline.

2.1. Knowledge Extraction and Lexicalization

We extract knowledge from three heterogeneous sources and convert it into a natural language format suitable for LLM consumption.

- **Textual Data**: Raw item descriptions (t_i) are used directly as they are already in text format.
- **Knowledge Graphs (KG)**: Structured KG triples (e.g., *(Tenet, director, Christopher Nolan)*) are converted into sentences using predefined templates (e.g., "*Tenet was directed by Christopher Nolan.*").
- **Collaborative Data**: We mine association rules from the user-item interaction matrix using the Apriori algorithm [1]. Rules (e.g., {item A} $\rightarrow$ {item B, C}) are lexicalized into sentences like, "*People who like item A also tend to like item B and item C.*"

This multi-source approach is inspired by KARS research showing that combining these data types yields robust performance [3].

2.2. Unified Training Phase

We adopt a unified training strategy that combines task adaptation with knowledge injection in a single step. The model is fine-tuned on a dataset containing two types of examples:

1. **Instruction-Tuning Data**: For each user, we generate prompts that frame the re-ranking task. The input contains the user's history and candidate items, and the target is the ground-truth ranked list.

Table 1
Main results comparing our method (LLaMA-KI) against baselines. LLaMA-KI uses the best knowledge configuration for each domain (Collaborative for Movies, Text for Music, All for Books). * indicates statistical significance ($p < 0.05$) over the best baseline.

Model	MovieLens 1M				Last.FM (Music)				DBbook (Books)			
	P@5 ↑	R@5 ↑	NDCG@5 ↑	Pop@5 ↓	P@5 ↑	R@5 ↑	NDCG@5 ↑	Pop@5 ↓	P@5 ↑	R@5 ↑	NDCG@5 ↑	Pop@5 ↓
Traditional Baselines												
BPR	0.7519	0.1860	0.7592	0.1617	0.6615	0.3313	0.6867	0.0701	0.5777	0.4672	0.6053	0.0166
LightGCN	0.7410	0.1811	0.7540	0.1491	0.6732	0.3381	0.7056	0.0659	0.5749	0.4639	0.6101	0.0163
LLM Baselines												
LLaMA 3 (zero-shot)	0.5615	0.1418	0.7130	0.1226	0.3861	0.2116	0.5340	0.0404	0.4630	0.3813	0.6369	0.0128
P5	0.6532	0.1602	0.6524	0.1077	0.5881	0.2920	0.5978	0.0597	0.5433	0.4413	0.5728	0.0115
GPT-4 (zero-shot)	**0.7683**	0.1820	**0.8132**	0.1443	0.5888	0.2880	0.6503	0.0455	0.5748	0.4590	0.6802	0.0120
Our Approach												
LLaMA (w/o knowledge)	0.7654	**0.2105**	0.7728	**0.0921**	0.8089	0.4272	0.8042	0.0390	0.7816	0.6317	0.8601	**0.0113**
LLaMA-KI (Ours)	0.7611	0.2070	0.7709	0.0927	**0.8428***	**0.4433***	**0.8490***	0.0392	**0.8122***	**0.6615***	**0.9015***	**0.0113***

2. **Knowledge-Tuning Data**: The lexicalized textual, KG, and collaborative information for each item is formatted as input-output pairs where the model learns to reconstruct this knowledge.

Training optimizes a total loss $L_{\text{total}} = L_{\text{k}} + L_{\text{i}}$, where L_{k} is the reconstruction loss for the knowledge data and L_{i} is the prediction loss for the instruction-tuning (re-ranking) data. Both are standard cross-entropy losses for next-token prediction. We use Low-Rank Adaptation (LoRA) [8] for parameter-efficient fine-tuning.

2.3. Inference

At inference time, the fine-tuned LLM is given a prompt containing a test user's history and a list of candidate items. The model generates a ranked list, which is parsed to extract the final top-k recommendations.

3. Experimental Setup

Our experiments address three research questions: (RQ1) How do individual knowledge types affect performance? (RQ2) How does combining knowledge sources impact performance? (RQ3) How does our model compare to state-of-the-art baselines?
Datasets. We use three public datasets: **MovieLens 1M** (movies), **Last.FM** (music), and **DBbook** (books). Item features (textual, graph) are mapped from DBpedia.
Implementation. We use **LLaMA 3 8B Instruct** as our base model. The evaluation protocol follows a standard user-based split (80% fine-tuning, 20% test).
Baselines. We compare our model against three families of baselines:

- **Collaborative Filtering**: BPR [13], MultiVAE [9], and SimpleX [10].
- **Graph-based**: LightGCN [6] and CFKG [2].
- **LLM-based**: Zero-shot prompting with **GPT-3.5**, **GPT-4**, the base **LLaMA 3** model, and the tuned **P5** model [5].

Metrics. We evaluate top-5 recommendations using **Precision@5**, **Recall@5**, **nDCG@5**, and average item **Popularity@5** (lower is better, indicating less reliance on popular items).

4. Results and Discussion

Our results are summarized in Table 1 and organized by our research questions.

4.1. RQ1 & RQ2: Impact of Knowledge

We first analyzed the effect of injecting different knowledge sources individually and in combination. The findings are highly domain-dependent.

For **MovieLens**, simply fine-tuning the LLM on the re-ranking task without any external knowledge ('LLaMA w/o knowledge') yielded the best results within our framework. Injecting additional knowledge did not provide further gains and in some cases slightly degraded performance. This suggests that the base LLaMA 3 model already possesses extensive knowledge about the popular movie domain, making additional injection redundant. The massive, proprietary GPT-4 model performs best overall on this dataset, likely due to its even larger scale and more comprehensive pre-trained knowledge base.

For **Last.FM (Music)**, the scenario is reversed. Here, injecting external knowledge provides a substantial and statistically significant performance boost. The best-performing single source was *Textual* data, which significantly outperformed the no-knowledge variant. Combining knowledge sources did not yield further improvements over using textual data alone. This indicates that for music, rich descriptive text is the most critical missing piece of information for the LLM.

For **DBbook (Books)**, knowledge injection was again highly effective. The best performance was achieved by combining all three knowledge sources (*Collaborative + Graph + Text*), which delivered a statistically significant improvement over the no-knowledge baseline. This suggests that the book domain benefits from a more holistic set of information, blending content descriptions, structured metadata, and user behavior patterns.

A key takeaway is that **the value of knowledge injection is inversely proportional to the domain's representation in the LLM's pre-training data**. For well-covered domains like movies, task-tuning is sufficient. For specialized or niche domains like music and books, explicit knowledge injection is crucial for achieving high accuracy.

4.2. RQ3: Comparison with Baselines

As shown in Table 1, our approach demonstrates state-of-the-art performance.

First, fine-tuning LLaMA 3 (even without knowledge) dramatically outperforms zero-shot LLM baselines (including GPT-4 in many cases) and traditional methods on the music and book datasets. This highlights the power of adapting an LLM to the specific task and data distribution.

Second, our final knowledge-injected model, **LLaMA-KI**, sets a new state of the art on the **music and book domains**, decisively outperforming all other models, including the much larger GPT-4. This is a critical finding: a smaller, open-source model, when infused with the right domain knowledge, can surpass a massive, general-purpose model. This shows that targeted knowledge is a more efficient path to high performance in specialized domains than simply scaling up the model size.

In the **movie** domain, while our tuned model significantly outperforms traditional baselines and other LLM approaches like P5, it does not surpass GPT-4 in terms of nDCG. However, it achieves higher recall and recommends less popular items, indicating a better ability to handle the cold-start setting of our evaluation.

5. Conclusion and Future Work

We presented a versatile pipeline for injecting multi-source domain knowledge into LLMs for recommendation. Our experiments demonstrate that this approach is highly effective, yielding state-of-the-art results, particularly in domains where a general-purpose LLM's pre-trained knowledge is insufficient. The results underscore a key principle: for specialized recommendation tasks, targeted knowledge injection can be more valuable than raw model scale.

Future work will focus on: (1) integrating more diverse knowledge sources like user reviews and multimodal data (e.g., images, audio); (2) developing methods to automatically assess the quality and relevance of knowledge sources before injection; and (3) exploring the trade-offs between model size, the amount of injected knowledge, and computational costs.

Acknowledgement

This research is partially funded by the PNRR project FAIR—Future AI Research (PE00000013), Spoke 6—Symbiotic AI, under the NRRP MUR program supported by NextGenerationEU (CUP H97G22000210007), and the PHaSE project — Promoting Healthy and Sustainable Eating through Interactive and Explainable AI Methods, funded by MUR under the PRIN 2022 program - Finanziato dall'Unione europea - NextGeneration EU, Missione 4 Componente 1 (CUP H53D23003530006).

The models are developed using the Leonardo supercomputer with the support of CINECA-Italian Supercomputing Resource Allocation, under class C projects IscrC_LLM-REC (HP10CTEUGX) and IscrC_SYMBREC (HP10C1A4P8).

Declaration on Generative AI

During the preparation of this work, the author did not use any AI tool.

References

[1] Rakesh Agrawal. Fast Algorithms for Mining Association Rules.

[2] 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.

[3] Vito Walter Anelli, Pierpaolo Basile, Tommaso Di Noia, Francesco Maria Donini, Antonio Ferrara, Cataldo Musto, Fedelucio Narducci, Azzurra Ragone, and Markus Zanker, editors. *Proceedings of the Sixth Knowledge-aware and Conversational Recommender Systems Workshop co-located with 18th ACM Conference on Recommender Systems (RecSys 2024), Bari, Italy, October 18th, 2024*, volume 3817 of *CEUR Workshop Proceedings*, 2024. CEUR-WS.org. URL https://ceur-ws.org/Vol-3817.

[4] Janneth Chicaiza and Priscila Valdiviezo-Diaz. A Comprehensive Survey of Knowledge Graph-Based Recommender Systems: Technologies, Development, and Contributions. *Information*, 12(6): 232, May 2021. ISSN 2078-2489. doi: 10.3390/info12060232. URL https://www.mdpi.com/2078-2489/12/6/232.

[5] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (RLP): A unified pretrain, personalized prompt & predict paradigm (P5). In Jennifer Golbeck, F. Maxwell Harper, Vanessa Murdock, Michael D. Ekstrand, Bracha Shapira, Justin Basilico, Keld T. Lundgaard, and Even Oldridge, editors, *RecSys '22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*, pages 299–315. ACM, 2022. doi: 10.1145/3523227.3546767. URL https://doi.org/10.1145/3523227.3546767.

[6] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu, editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 639–648. ACM, 2020. doi: 10.1145/3397271.3401063. URL https://doi.org/10.1145/3397271.3401063.

[7] Jonathan L. Herlocker, Joseph A. Konstan, Al Borchers, and John Riedl. An algorithmic framework for performing collaborative filtering. *SIGIR Forum*, 51(2):227–234, 2017. doi: 10.1145/3130348.3130372. URL https://doi.org/10.1145/3130348.3130372.

[8] 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. *arXiv preprint arXiv:2106.09685*, 2021.

[9] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. Learning disentangled representations for recommendation. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information*

Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pages 5712–5723, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/a2186aa7c086b46ad4e8bf81e2a3a19b-Abstract.html.

[10] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. Simplex: A simple and strong baseline for collaborative filtering. In Gianluca Demartini, Guido Zuccon, J. Shane Culpepper, Zi Huang, and Hanghang Tong, editors, *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 1243–1252. ACM, 2021. doi: 10.1145/3459637.3482297. URL https://doi.org/10.1145/3459637.3482297.

[11] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

[12] Alessandro Petruzzelli, Cataldo Musto, Marco de Gemmis, Giovanni Semeraro, and Pasquale Lops. Empowering recommender systems based on large language models through knowledge injection techniques. In *Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization*, UMAP '25, page 40–50, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713132. doi: 10.1145/3699682.3728341. URL https://doi.org/10.1145/3699682.3728341.

[13] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: bayesian personalized ranking from implicit feedback. In Jeff A. Bilmes and Andrew Y. Ng, editors, *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, pages 452–461. AUAI Press, 2009. URL https://www.auai.org/uai2009/papers/UAI2009_0139_48141db02b9f0b02bc7158819ebfa2c7.pdf.

[14] Hugo Touvron, Louis Martin, and Kevin Stone. Llama 2: Open Foundation and Fine-Tuned Chat Models.

[15] Lei Wang and Ee-Peng Lim. Zero-shot next-item recommendation using large pretrained language models. *CoRR*, abs/2304.03153, 2023. doi: 10.48550/ARXIV.2304.03153. URL https://doi.org/10.48550/arXiv.2304.03153.