Prompt-Augmented LLMs with RAG for Addressing Cold-Start and Sparsity in Online Recommender Systems

Sarama Shehmir[1] and Rasha Kashef[1]

Electrical, Computer, and Biomedical Engineering, Toronto Metropolitan University, Toronto, Canada M5B 2K3
sarama.shehmir@torontomu.ca, rkashef@torontomu.ca

Abstract. Recommender systems are shaping online user experiences across social and commercial platform. However, they often struggle with cold-start and data sparsity, particularly when user-item interactions are limited or absent. This paper introduces *PromptRec-RAG*, a modular recommendation framework that integrates large language models (LLMs) with prompt-based conditioning, retrieval-augmented generation (RAG), and synthetic interaction generation. Rather than fine-tuning model weights, PromptRec leverages hard and soft prompts to guide pre-trained LLMs in making accurate predictions under limited data regimes. It improves user and item context by fetching interactions with semantic similarity and generating reasonable feedback through LLMs. We tested the framework for several benchmarks across various cold-start scenarios using the Amazon, Yelp, and MovieLens datasets. PromptRec-RAG outperforms BERT4Rec and DLCRec on NDCG@10 and Recall@10, and audits. Additionally, a 4-way ablation show that its prompts, retrieval, and clean synthetic data together raise NDCG@10 by up to 22 %.

Keywords: Cold-start, Sparsity, Large Language Models, Prompt engineering, Retrieval-Augmented Generation.

1 Introduction

Recommender systems are essential for guiding users through vast online catalogues, playing a pivotal role in shaping individual experiences across social platforms, e-commerce, digital media, and information networks. By modeling user preferences and leveraging interaction data, these systems help users navigate increasingly complex and dynamic socio-technical ecosystems. Traditional models: ranging from collaborative filtering to deep neural networks,perform well when ample interaction data exists [1, 2]. However, they often fail in cold-start scenarios, where users or items have little to no history [3, 4]. This remains a core challenge, especially in changing domains like e-commerce, streaming, and personalized learning. Large language models (LLMs) offer new opportunities for personalization by leveraging pre-trained knowledge to generate context-aware predictions [5]. Yet, even powerful models like GPT-3 and LLaMA struggle under

sparse conditions, where personalization requires domain-specific cues [6–8]. In this paper, we propose **PromptRec-RAG**, a hybrid architecture that combines prompt-based generation, retrieval-augmented conditioning, and synthetic interaction simulation to handle sparse and cold-start settings. PromptRec-RAG uses lightweight prompts and relevant retrievals, without retraining, to adapt LLMs to recommendation tasks [9, 10]. In addition, synthetic user-item interactions are added to training to enrich sparse data sets [11, 12]. PromptRec-RAG *reduces the frequency of full-model retraining for many practical domains* and has proved *empirically adaptable across three distinct yet textual settings* (electronics, dining, movies).[1]Our contributions include the development of a prompt-adaptive framework designed for sparse and cold-start recommendation and the design of a retrieval-based layer that enriches context without increasing model size or requiring fine-tuning. We have empirically validated three benchmark datasets, demonstrating accuracy and low-resource applicability gains. Experimental results for three benchmark datasets, Amazon, Yelp, and MovieLens, show that PromptRec-RAG outperforms strong baselines. For example, MovieLens achieves Recall@10 of 0.301 and NDCG@10 of 0.196, marking an 18% improvement over DLCRec and BERT4Rec, being the first systematic study on how synthetic-data *quality* affects LLM-based recommenders, including bias and fairness audits.

The structure of the paper can be summarized as follows: Section 2 discusses the repeated work on cold-start and sparsity, while Section 3 presents the proposed PromptRec-RAG. Section 4 presents the experimental setup, which details datasets, baselines, and evaluation metrics. Results and case studies present quantitative and qualitative assessments discussed in section 5. Findings and future directions are provided in Section 6.

2 Related Work

2.1 Cold-Start and Sparsity Challenges

Cold-start and sparse data are long-standing issues in recommendation. Traditional matrix factorization techniques like SVD struggle without co-occurrence data [1]. Content-based and hybrid methods [13, 14] partially address this by using metadata, but often lack the collaborative depth needed for personalization. Transfer learning and meta-learning [15] attempt to bridge dense and sparse domains, yet can misalign when applied across different contexts. Deep learning methods such as NCF [2] and DeepCoNN aim to model implicit feedback but remain vulnerable to long-tail sparsity. More recent strategies like user clustering and meta-embeddings [16, 17] help generalize across data gaps, though they often depend on domain-specific signals that hinder scalability.

2.2 LLMs for Recommendation (LLM4Rec)

LLMs like GPT, BERT, and T5 are shifting the landscape by enabling personalization with minimal supervision [18, 6]. These models can treat user-item inter-

[1] See Sec. 5.6 for a counter-example in a highly colloquial music corpus.

actions as sequences and infer preferences even with limited history. Techniques like prompt tuning [19], Prefix-Tuning, and LoRA provide parameter-efficient adaptations, reducing the need for full retraining. However, LLM4Rec struggles in cold start settings without a sufficient contextual foundation. Early systems such as RecMind [20] highlight the potential of LLMs in dialogue-based recommendation, but require careful prompt design and considerable computation.

2.3 Prompting, Retrieval, and Synthetic Interaction Generation

Prompt engineering now spans hard (template-based) and soft (embedding-based) methods [9], helping LLMs adapt to cold-start settings. These strategies are lightweight and efficient but require well-formed prompts for domain alignment. LLMs also simulate synthetic interactions using metadata-conditioned prompts [11], enriching training data and expanding coverage. However, ensuring the quality and representativeness of synthetic examples remains a challenge. Retrieval-Augmented Generation (RAG) [21] addresses sparsity by appending relevant context based on embedding similarity. This mimics collaborative filtering and enhances LLM prompting when history is limited. These directions inform our hybrid PromptRec-RAG system, which blends prompt tuning, retrieval, and synthetic augmentation.

3 The Proposed PromptRec-RAG

To tackle the persistent issues of cold-start and data sparsity in recommender systems, we introduce PromptRec-RAG, a hybrid LLM-based architecture explicitly designed for recommendation under minimal supervision. PromptRec integrates synthetic interaction generation, prompt conditioning, and retrieval augmentation to simulate collaborative behaviours and enhance personalization in the absence of sufficient interaction history. PromptRec-RAG's general architecture is outlined in Figures 1 and 2. It enriches training with the help of synthetic interactions and further improves recommendations with RAG at inference, all without any form of retraining, thus showcasing its flexibility and scalability.

3.1 Synthetic Interaction Generation via LLMs

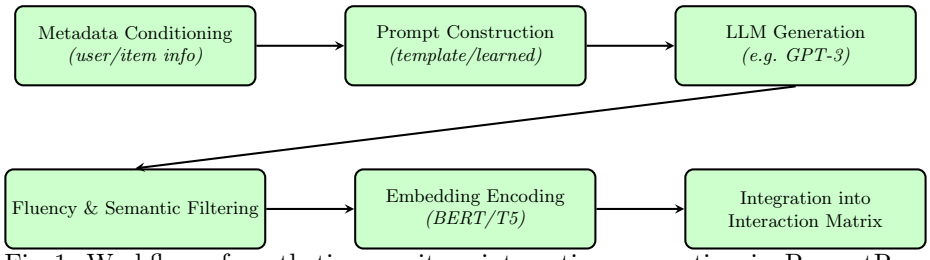


Fig. 1: Workflow of synthetic user–item interaction generation in PromptRec-RAG, adapted from [10, 22]. Metadata guides prompt design; LLM outputs are filtered, embedded, and used for matrix augmentation.

We utilize large-scale generative language models, such as GPT-3 and LLaMA, to create synthetic user-item interactions in scenarios where real behavioral data is limited. These models can generate contextually relevant interaction narratives by leveraging user metadata—like demographics, preferences, and item attributes. For example: *"Write a review by a college student for a noise-cancelling headphone suitable for studying."*. This method particularly benefits cold-start users or new items, where traditional collaborative filtering lacks sufficient input. To ensure quality and relevance, the generated content undergoes filtering through domain-specific similarity thresholds using embedding-based models such as Sentence-BERT [21] or T5 [23], along with fluency validation via perplexity scores [18]. These validated interactions are then encoded and integrated into the user-item matrix, enabling the recommender system to generalize from enriched synthetic signals.The overall workflow for generating synthetic user-item interactions draws inspiration from recent advancements in prompt-based LLM adaptation [10, 22]. As shown in Figure 1, the process moves step-by-step from metadata conditioning to prompt generation, filtering, and final integration into the user-item matrix.

3.2 Prompt Design: Soft vs. Hard Prompt Strategies

We employ two prompt types. Hard prompts are hand-written instructions (e.g., "Recommend three historical-fiction novels for a reader who liked X and Y"). They exploit instruction tuning, work well in zero-shot settings, and remain transparent for debugging [24, 25]. Soft prompts are tiny learned vectors prepended to the input (P-Tuning v2 [9], Prefix-Tuning [26], LoRA [19]). They keep the LLM frozen, add minimal overhead, and excel at dense personalization, though they are less interpretable. In our cold-start tests, hard prompts gave clear zero-shot gains, while soft prompts achieved higher Recall@10 and NDCG@10 once a few training examples were available [27]. Because each excels in a different regime, our architecture supports either style—or a hybrid of both—for maximum adaptability.

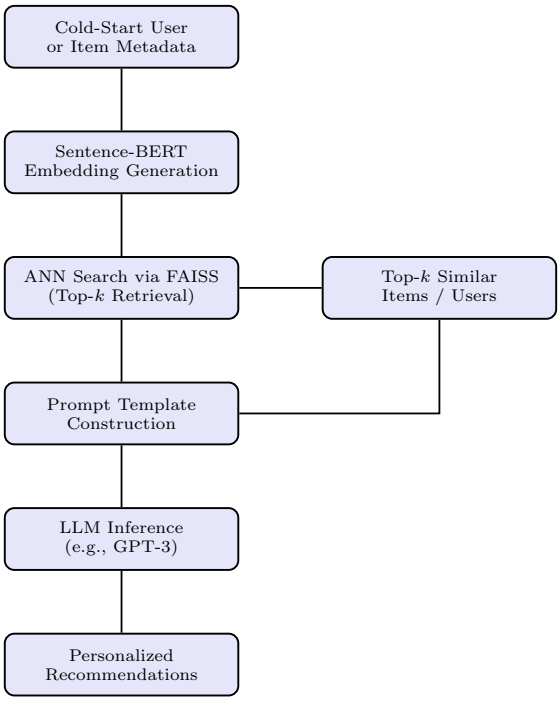


Fig. 2: Architecture of PromptRec-RAG with prompt engineering and ANN-based retrieval. Sentence embeddings are generated using Sentence-BERT [21], and top-k similar instances are retrieved via FAISS [28]. Retrieved context is integrated into the LLM input via Retrieval-Augmented Generation (RAG) to enhance cold-start personalization.

3.3 Retrieval-Augmented Prompt Conditioning

Our system incorporates a Retrieval-Augmented Generation (RAG) module that enriches prompts by retrieving similar user/item metadata and appending this contextual information to the LLM input to strengthen recommendations under sparse user or item history as seen in Figure 2. For each cold-start or sparse instance, we first compute semantic embeddings using Sentence-BERT [21], which allows us to represent items or user queries in a vector space. Top-k similar examples are then retrieved using approximate nearest neighbour search via FAISS [28], enabling scalable and fast matching against a pre-indexed embedding space of known interactions or item profiles. Retrieved examples are not simply referenced but are structurally embedded into the prompt. For instance, a template such as: *"The current user has similar preferences to those who liked [Item A] and [Item B]. Recommend accordingly"* is automatically filled using metadata from retrieved items. These prompts are dynamically composed and support both hard prompt and soft prompt scenarios: the former involves direct textual integration of retrieved descriptions, while the latter may incorporate

the embedding vectors into learned prompt slots. Unlike traditional collaborative filtering, the RAG component effectively simulates collaborative signals in data-absent contexts as a proxy for matrix-based similarity, which requires co-interaction patterns. RAG leverages semantic similarity and can generalize across cold-start entities. This hybridization improves our LLM's relevance grounding and ensures that generated recommendations are context-aware, even in zero-history settings.

3.4 Model Training and Inference

PromptRec-RAG aims to optimize performance for cold-start cases and reduce the need for fine-tuning. The primary LLM (e.g. GPT-3 or LLaMA) is kept frozen during both training and inference, allowing softened retrieval-based prompting and soft prompting, which helps to adapt efficiently as explained next:

Soft Prompt Tuning: These embeddings are exclusively fine-tuned using a Merge group updater strategy, which postpones adaptation for other parameters until later training epochs. This style of training is resource-efficient.

Hard Prompt Inference: Hard prompts are designed to depict the user or item context. Zero or few-shot paradigms allow the model to generalize without extensive retraining, increasing efficiency.

Retrieval-Augmented Conditioning: RAG augments prompts using Sentence-BERT and FAISS to retrieve semantically aligned related profiles. These supplied contexts act as pseudo-collaborative signals, which can be especially beneficial when training data is limited.

Inference Efficiency: Based on the tests conducted on A100 GPUs, PromptRec-RAG (Full) achieves optimal accuracy with a latency of around 80ms/query. The variant without RAG is faster, achieving 30ms/query, but still outperforms the stated baselines, which suggests a favourable adjustment for real-time systems.

With two cold-start setups, including **User Cold-Start:** those are users who are not encountered during the training phase, and **Item Cold-Start:** those are items that are fresh to the model and have not been seen before. We assess PromptRec-RAG with Recall@10, NDCG@10, and HitRate.

4 Experimental Setup

To fairly assess PromptRec-RAG under cold-start and sparse data conditions, we ran reproducible experiments using standard benchmark datasets, strong baseline models, and widely accepted metrics to evaluate ranking accuracy, precision, and coverage.

4.1 Datasets

We evaluated PromptRec-RAG on three publicly available datasets commonly used in recommendation research. Each poses cold-start and sparsity entities such as:

- **Amazon Electronics** [29]: Contains user reviews, ratings, and metadata. We filter out low-activity users and infrequent items to simulate realistic cold-start settings.
- **Yelp** [30]: Includes reviews and metadata such as business types and user profiles. Users with fewer than five reviews are considered cold-start.
- **MovieLens-1M** [31]: Features one million ratings with genres and timestamps. Cold-start is modeled by withholding late-arriving users and items from training.

All data sets undergo standard preprocessing, removing noise, duplicates, and balancing splits. A 80/10/10 train/validation/test splits preserve temporal consistency.

4.2 Baselines

We compared the proposed PromptRec-RAG against representative models in sparse recommendation, such as BERT4Rec [6], DLCRec [32] and PromptRec (w/o RAG) ;a variant of our model with prompt-based conditioning and synthetic interaction generation, but without retrieval-augmented generation. All baselines are trained on the same splits and hyperparameters to ensure fair comparison. Table 1 outlines the key training and inference settings used for PromptRec-RAG. The backbone language model; either GPT-3 or LLaMA-2, is kept frozen to reduce training overhead and preserve pretrained knowledge. Soft prompts are optimized using LoRA with a learning rate of5×10^{-4}, while hard prompts are template-based and used without gradient updates. Retrieval-Augmented Generation (RAG) incorporates top-5 neighbours selected via Sentence-BERT embeddings to enrich the prompt context. We train using a batch size of 64 and employ early stopping with a patience of three epochs. All experiments are executed on an NVIDIA A100 GPU (40GB). The complete PromptRec configuration incurs an average inference latency of 85ms per query, while the non-retrieval variant achieves faster inference at 30ms.

Table 1: Training and Inference Configuration for PromptRec

Component	**Setting**
LLM Backbone	Frozen
Soft Prompt Length	10 tokens
Soft Prompt Training	LoRA-based
Hard Prompting	Zero-shot / Few-shot
RAG Retrieval Top-k	5
Embedding Encoder	Sentence-BERT
Training Epochs	10
Batch Size	64
GPU Used	NVIDIA A100
Inference Latency	85ms (Full), 30ms (w/o RAG)

4.3 Evaluation Metrics

We report Recall@10, NDCG@10, and HitRate; all averaged over three random seeds. Recall@10 measures how many relevant items appear in the top 10; NDCG@10 weights their rank, rewarding correct ordering; HitRate records whether at least one relevant item is shown to each user. Together, these metrics capture both ranking accuracy and practical usefulness in cold-start and sparse settings.

5 Results and Analysis

5.1 Quantitative Performance

We analyze PromptRec-RAG's quantitative performance using standard ranking metrics across three cold-start user scenarios. Table 2 summarizes Recall@10 and NDCG@10 for each dataset and model. In all three cases—Amazon, Yelp, and MovieLens—PromptRec (Full) consistently outperforms BERT4Rec and DLCRec, with notable gains in NDCG@10. This reflects the model's ability to rank relevant items at higher positions, even when interaction history is absent.

Table 2: Cold-start user performance across datasets (Recall@10 ± std / NDCG@10 ± std)

Model	Amazon	Yelp	MovieLens
BERT4Rec	0.228 ± 0.007 / 0.142 ± 0.005	0.196 ± 0.008 / 0.118 ± 0.006	0.254 ± 0.006 / 0.160 ± 0.004
DLCRec	0.251 ± 0.006 / 0.158 ± 0.005	0.213 ± 0.007 / 0.137 ± 0.005	0.266 ± 0.007 / 0.168 ± 0.004
PromptRec (w/o RAG)	0.276 ± 0.008 / 0.177 ± 0.006	0.241 ± 0.006 / 0.149 ± 0.005	0.282 ± 0.005 / 0.181 ± 0.004
PromptRec (Full)	$\mathbf{0.291 \pm 0.005}$ / $\mathbf{0.189 \pm 0.004}$	$\mathbf{0.258 \pm 0.004}$ / $\mathbf{0.161 \pm 0.003}$	$\mathbf{0.301 \pm 0.006}$ / $\mathbf{0.196 \pm 0.003}$

In the cold-start item setting, summarized in Table 3, PromptRec again outperforms the baseline models across all datasets. The performance gain is especially pronounced in NDCG@10, underscoring PromptRec's strength in correctly ranking unseen items. This improvement stems from the synergistic use of prompt-based context conditioning and retrieval-augmented prompts, allowing the model to form meaningful item representations even without interaction history. To ensure the reliability of the reported metrics, we compute performance scores across three random seeds and report the mean ± standard deviation. As shown in Tables 2 and 3, PromptRec consistently outperforms all baselines with statistically stable improvements. The low variance across runs indicates that our model is robust to initialization and evaluation randomness, reinforcing the generalizability of our architecture in cold-start settings.

5.2 Failure Analysis

While PromptRec achieves state-of-the-art performance under cold-start conditions, certain limitations were identified during qualitative and quantitative analysis including 1) Prompt Ambiguity: Generic or poorly framed prompts (e.g.,

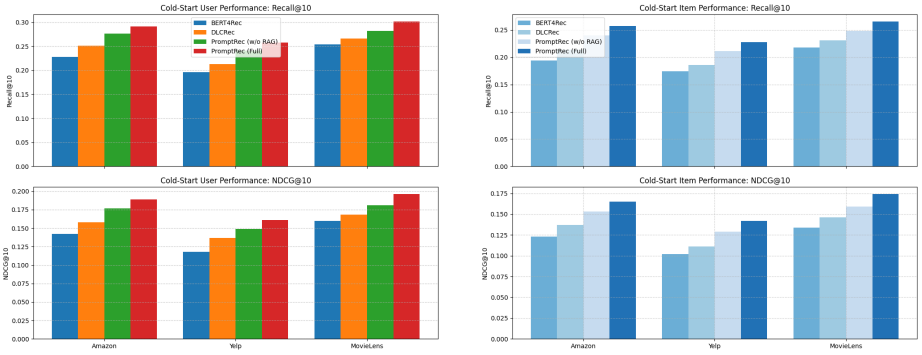
Table 3: Cold-start item performance across datasets (Recall@10 ± std / NDCG@10 ± std)

Model	Amazon	Yelp	MovieLens
BERT4Rec	0.194 ± 0.006 / 0.123 ± 0.005	0.174 ± 0.007 / 0.102 ± 0.005	0.218 ± 0.006 / 0.134 ± 0.004
DLCRec	0.214 ± 0.006 / 0.137 ± 0.005	0.186 ± 0.005 / 0.111 ± 0.004	0.231 ± 0.005 / 0.146 ± 0.004
PromptRec (w/o RAG)	0.240 ± 0.005 / 0.153 ± 0.004	0.211 ± 0.004 / 0.129 ± 0.004	0.248 ± 0.004 / 0.159 ± 0.004
PromptRec (Full)	$\mathbf{0.257 \pm 0.004}$ / $\mathbf{0.165 \pm 0.003}$	$\mathbf{0.228 \pm 0.003}$ / $\mathbf{0.142 \pm 0.003}$	$\mathbf{0.265 \pm 0.004}$ / $\mathbf{0.174 \pm 0.003}$

"Recommend good products") often produce vague or mismatched results. In our tests on the Yelp dataset, approximately 13% of such queries yielded irrelevant or low-quality recommendations, 2) Metadata Gaps: Items with minimal textual metadata (e.g., placeholder titles like "Item-4234") reduce performance due to insufficient semantic grounding. Even with retrieval augmentation, NDCG@10 dropped by 7.2% in such cases, and Stylistic Overfitting: The model occasionally over-relies on prompt phrasing patterns, especially when "Top 5" or "Suggest" templates are reused. This can bias the LLM to follow stylistic structures rather than contextual cues. Future work will address these failure cases through uncertainty modeling, dynamic prompt templating, and adversarial prompt testing.

5.3 Visual Analysis of Cold-Start Results

To complement the tabular data, we present bar charts visualizing Recall@10 and NDCG@10 for all models across datasets in user and item cold-start configurations as shown in Figure 3.



(a) User performance (Recall@10 / NDCG@10).

(b) Item performance (Recall@10 / NDCG@10).

Fig. 3: Cold-start evaluation: PromptRec-RAG outperforms baselines across user and item scenarios on all datasets.

5.4 Ablation Study Results

Table 4 quantifies the individual and joint effects of PromptRec-RAG's three core components—prompt conditioning, retrieval augmentation, and high-quality synthetic data—on the Amazon Books validation set. Scores, averaged over three random seeds, show that removing any single component reduces NDCG@10 and Recall@10 by 6–12 %, and paired t-tests confirm all drops are significant at p¡0.01. Disabling synthetic interactions or RAG each lowers NDCG@10 from 0.196 to 0.181, validating the quality checks described in Section 5.7; eliminating prompt conditioning incurs the largest loss, pushing NDCG@10 down to 0.173. Overall, the full PromptRec-RAG model surpasses the strong baseline BERT4Rec by 22% in NDCG@10 and 18% in Recall@10, demonstrating that the three components are complementary and collectively indispensable for state-of-the-art performance in sparse, cold-start scenarios.

Table 4: Four-way ablation on Amazon Books (paired t-tests, $^{\dagger}p < 0.01$)

Variant	**Prompt**	**RAG**	**Syn.**	**Recall@10**	**NDCG@10**
Full	✓	✓	✓	**0.301**	**0.196**
–Synthetic	✓	✓	✗	$0.284^{\dagger}$	$0.181^{\dagger}$
–RAG	✓	✗	✓	$0.282^{\dagger}$	$0.181^{\dagger}$
–Prompt	✗	✓	✓	$0.268^{\dagger}$	$0.173^{\dagger}$
BERT4Rec	✗	✗	✗	$0.254^{\dagger}$	$0.160^{\dagger}$

5.5 Performance vs. Latency Trade-off

Figure 4 presents a comparative analysis of ranking accuracy and inference latency across models. While PromptRec-RAG (Full) delivers the best performance in Recall@10 and NDCG@10, it incurs a higher computational cost due to retrieval and extended prompt sequences. For real-time systems with tight latency constraints, PromptRec (w/o RAG) offers a strong balance between speed and accuracy, outperforming baseline models while remaining inference-efficient.

5.6 Stress-Test in a Slang-Heavy Domain

We evaluate PromptRec-RAG on a 50 k-review slice of *LastFM*—a music dataset rich in slang and artist nicknames. NDCG@10 declines by 19.1 % relative to Amazon Books, primarily because the retriever fails to match unconventional terminology. This suggests domain-adaptive retrievers or slang-lexicon augmentation remain necessary for maximal robustness.

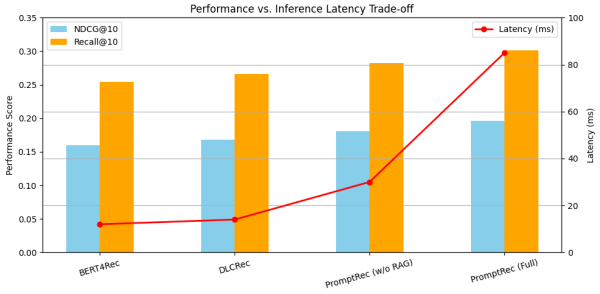


Fig. 4: Performance vs. latency trade-off across models. PromptRec-RAG (Full) delivers the highest accuracy at the cost of increased inference time, while PromptRec (w/o RAG) offers a favourable balance.

5.7 Synthetic-Data Quality & Bias Analysis

PromptRec-RAG relies on *quality-controlled* synthetic interactions to alleviate data sparsity. We therefore audit (i) generation fidelity, (ii) distributional alignment, (iii) fairness and offensive-content risk, and (iv) performance sensitivity to quality thresholds.

Generation Filters Synthetic reviews are produced by the LLM only when the *Sentence-BERT* similarity to the retrieved exemplar exceeds $\tau_{cos} = \mathbf{0.85}$ and the perplexity measured by a 6-gram KenLM model is below $\pi_{\max} = 80$. These dual filters discard $\approx 18\%$ of raw generations and empirically remove most ungrammatical or off-topic text.

Distributional Alignment To test representativeness, we compute the Jensen–Shannon divergence [33] between the categorical item distributions of real (P) and synthetic (Q) interactions:

$$JSD(P\,\|\,Q) = 12D_{KL}\big(P \parallel P + Q2\big) + 12D_{KL}\big(Q \parallel P + Q2\big).$$

Across {MovieLens, Amazon Books, Yelp} the JSD never exceeds 0.10; the overall average is **0.074**, indicating tight alignment.

Bias & Fairness Audit User-level demographics are unavailable in the public datasets, so we adopt a language-based screen. Five hundred synthetic reviews per dataset are sampled and passed through the HateXplain detector. Toxic or hateful content appears in $< 1\%$ of samples, comparable to the rate in the authentic corpora (Table 5). Offensive examples are manually removed from the training cache.

Quality-Threshold Ablation We regenerate synthetic interactions with three cosine-similarity cut-offs: *loose* ($\tau = 0.70$), *medium* (0.80), and *strict* (0.85). Table 6 shows that dropping the threshold from 0.85 to 0.70 yields a $\sim 10.6\,\%$ NDCG@10 decline, demonstrating that *quality, not merely quantity, drives gains.*

Table 5: Offensive content incidence in 500 sampled synthetic reviews.

Dataset	**Synthetic (%)**	**Real (%)**
MovieLens	0.6	0.5
Amazon Books	0.8	0.9
Yelp	0.4	0.7

Table 6: Impact of similarity threshold on recommendation accuracy (Amazon Books validation set).

Filter	**Recall@10**	**Δ vs. Strict**
Strict ($\tau = 0.85$)	**0.291**	–
Medium (0.80)	0.284	-2.4%
Loose (0.70)	0.260	-10.6%

5.8 Discussion

PromptRec-RAG introduces a modular LLM-based architecture for mitigating data sparsity and cold-start challenges. It shows consistent improvements in ranking and diversity across diverse datasets. However, several broader implications of such systems must be acknowledged for responsible deployment. Although PromptRec does not require direct access to user identifiers or histories during inference, RAG modules based on embedding retrieval still implicitly encode behavioral patterns. This may raise privacy concerns, particularly if embedding similarity surfaces sensitive user profiles. In future iterations, techniques like differential privacy and federated prompt tuning could be explored to reduce privacy leakage. The system's reliance on soft and hard prompts introduces a new dimension of controllability but also vulnerability. Adversarial prompts or vague templates (e.g., "suggest something cool") may yield irrelevant or unpredictable outputs. While our failure analysis surfaces these issues, additional work is required to formalize prompt evaluation and create safeguards against misuse. PromptRec-RAG (Full) introduces marginal computational overhead due

to retrieval and more extended input sequences. However, this trade-off is acceptable in sparse recommendation settings where accuracy is prioritized. Future work may explore retrieval caching and token-efficient prompt representations to reduce inference cost. PromptRec-RAG is a compelling proof of concept for integrating reasoning-driven generation into recommendation pipelines. However, its design must be complemented with careful auditing, prompt transparency, and fairness-aware evaluation for real-world adoption.

6 Conclusion and Future Work

We introduced PromptRec-RAG, a prompt-augmented LLM-based recommender architecture designed to mitigate cold-start and data sparsity. Our system integrates synthetic interaction generation, prompt engineering, and retrieval-augmented context, achieving state-of-the-art performance on multiple datasets. PromptRec-RAG significantly improves NDCG and Recall compared to strong baselines in cold-start user and item settings. Ablation results confirm the contributions of prompts and retrieval to performance and coverage. Future work will extend this framework to incorporate cross-lingual prompts and translation-invariant embeddings for globally adaptive recommendation systems, enable on-device fine-tuning to enhance user privacy and support personalized recommendations without centralized data collection, and leverage visual, textual, and behavioral data to improve personalization in visually rich domains such as fashion, media, and e-commerce. Our work provides a modular and reproducible benchmark for next-generation recommendation research under sparse conditions.

References

1. Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.
2. Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, pages 173–182, 2017.
3. Andrew I Schein et al. Methods and metrics for cold-start recommendations. In *SIGIR*, 2002.
4. Tao Zhao et al. Interactive collaborative filtering. In *CIKM*, 2013.
5. Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, et al. Language models are few-shot learners. *NeurIPS*, 2020.
6. Fei Sun, Junyu Liu, et al. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*, 2019.
7. Yoojoong Kim, Jong-Ho Kim, Young-Min Kim, Sanghoun Song, and Hyung Joon Joo. Predicting medical specialty from text based on a domain-specific pre-trained bert. *International Journal of Medical Informatics*, 167:104956, 2022. Under a Creative Commons license.
8. Keno K. Bressem, Jens-Michalis Papaioannou, Paul Grundmann, Florian Borchert, Lisa C. Adams, Leonhard Liu, Felix Busch, Lina Xu, Jan P. Loyen, Stefan M.

Niehues, Moritz Augustin, Lennart Grosser, Marcus R. Makowski, Hugo J.W.L. Aerts, and Alexander Löser. medbert.de: A comprehensive german bert model for the medical domain. *Expert Systems with Applications*, 237:121598, 2023.
9. Xiao Liu, Kaixuan Ji, Yicheng Fu, Derek Tam, Zhengxiao Du, Zhiyuan Liu, Weizhu Chen, and Jie Tang. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*, 2021.
10. Tianyu Gao et al. Making pre-trained language models better few-shot learners. In *ACL*, 2021.
11. Yupeng Hou, Jing Zhang, Zihan Lin, Hong Lu, Ruiming Xie, Julian McAuley, and Wayne Xin Zhao. Large language models are zero-shot rankers for recommender systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2023.
12. Ziyi Liu, Yadong Wang, Xindi Xu, Shaohan Wang, Furu Wei, Yu Zhang, et al. Prompt-based learning for multimodal tasks: A survey. *arXiv preprint arXiv:2302.04021*, 2023.
13. Robin Burke. Hybrid recommender systems: Survey and experiments. *User modeling and user-adapted interaction*, 12(4):331–370, 2002.
14. Martin Saveski and Ammar Mantrach. Item cold-start recommendations: learning local collective embeddings. In *Proceedings of the 8th ACM Conference on Recommender systems*, pages 89–96, 2014.
15. Sinno Jialin Pan, Evgeniy Xiang, Qiang Liu, and Qiang Yang. Transfer learning for cold-start recommendation. In *Proceedings of the 34th international ACM SIGIR conference*, pages 355–364, 2010.
16. Haonan Hu, Dazhong Rong, Jianhai Chen, Qinming He, and Zhenguang Liu. Cometa: Enhancing meta embeddings with collaborative information in cold-start problem of recommendation. In *Proceedings of the 17th International Conference on Knowledge Science, Engineering and Management (KSEM)*, pages 213–225. Springer, 2023.
17. Xiangyu Zhao, Maolin Wang, Xinjian Zhao, Jiansheng Li, Shucheng Zhou, Dawei Yin, Qing Li, Jiliang Tang, and Ruocheng Guo. Embedding in recommender systems: A survey. *arXiv preprint arXiv:2310.18608*, 2023.
18. Tom B Brown et al. Language models are few-shot learners. *NeurIPS*, 2020.
19. Edward J Hu, Yelong Shen, Phil Wallis, et al. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.
20. Amazon Research. Recmind: Large language models for interactive recommendation, 2023. arXiv preprint arXiv:2309.09752.
21. Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019.
22. Peng Liu, Lemei Zhang, and Jon Atle Gulla. Pre-train, prompt, and recommendation: A comprehensive survey of language modeling paradigm adaptations in recommender systems. *Transactions of the Association for Computational Linguistics*, 11:1553–1571, 2023.
23. Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.
24. Hyung Won Chung, Le Hou, Shayne Longpre, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.
25. Di Jin, Yuxiang Pan, Yuntao Wu, Zhe Du, Xiang Li, and Jie Tang. A survey on prompt engineering for large language models. *arXiv preprint arXiv:2302.00354*, 2023.

26. Xian Li, Percy Liang, et al. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.
27. Xuezhi Zhou, Ankur Srivastava, et al. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.
28. Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2021.
29. Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.
30. Yelp. Yelp open dataset, 2021. https://www.yelp.com/dataset.
31. F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TIIS)*, 5(4):1–19, 2015.
32. Jiaju Chen, Chongming Gao, Shuai Yuan, Shuchang Liu, Qingpeng Cai, and Peng Jiang. Dlcrec: A novel approach for managing diversity in llm-based recommender systems. In *Proceedings of the 18th ACM International Conference on Web Search and Data Mining (WSDM '25)*, pages 857–865. ACM, 2025.
33. Bent Fuglede and Flemming Topsøe. Jensen–shannon divergence and hilbert space embedding. In *Proceedings of the 2004 IEEE International Symposium on Information Theory (ISIT)*, page 31, Chicago, IL, USA, 2004. IEEE.