Conversational Recommenders via Repeated Sampling: Extending the Paradigm of the LLMonkeys

Joaquín Peralta
jperaltaperez@uc.cl
Pontificia Universidad Católica de Chile
Santiago, Chile

Nicolás Salazar
nicosalazar@uc.cl
Pontificia Universidad Católica de Chile
Santiago, Chile

Tomás Trincado
ttrincadoc@gmail.com
Pontificia Universidad Católica de Chile
Santiago, Chile

Abstract

Large Language Models (LLMs) have become pivotal in the development of conversational recommender systems, enabling rich, personalized interactions. However, most existing systems rely on a single inference per prompt, which can limit performance, particularly when using lightweight models. This project investigates whether repeated sampling (generating multiple responses from a lightweight LLM for a single user query) can enhance recommendation quality. Using the LLM-REDIAL dataset, we implemented and evaluated multiple open-source models (e.g., TinyLlama 1.1B, Gemma 2B, Falcon RW 1B), comparing them against high-capacity commercial models (GPT-4.1-nano, GPT-4.1-mini and GPT-4.1). Our experiments include Zero-Shot, Few-Shot, and Fine-Tuned settings with user interaction history. Additionally, we included metrics for Uncertainty, Diversity and Novelty to assess the model responses. Our results indicate that repeated sampling significantly improves recommendation quality, and may allow smaller models to approach or even surpass the performance of larger models when only single inferences are allowed. These findings suggest that sampling-based techniques can serve as a cost-effective strategy to improve conversational recommender systems.

CCS Concepts

• **Computing methodologies** → **Natural language generation**.

Keywords

Recommendation, Redial, LLM, LLMonkeys, TinyLlama, Gemma, Repeated, Sampling, Conversational, Recommenders

ACM Reference Format:
Joaquín Peralta, Nicolás Salazar, and Tomás Trincado. 2025. Conversational Recommenders via Repeated Sampling: Extending the Paradigm of the LLMonkeys. In . ACM, New York, NY, USA, 6 pages.

1 Introduction

Large language models (LLMs) have significantly advanced the quality of natural language understanding and generation across tasks such as dialogue, summarization, and recommendation. While most improvements have been driven by scaling model size and dataset quality, relatively less attention has been paid to the inference-time strategies that determine how these models are deployed in real-world applications.

Recent work has demonstrated that repeated sampling—i.e., issuing multiple inference attempts with stochastic decoding—can dramatically increase task success in settings such as code generation and theorem proving [1]. In these domains, a simple strategy of generating multiple candidate outputs and selecting the best via automated verification has led to substantial gains in performance, even allowing smaller models to outperform state-of-the-art systems with single-shot inference.

Inspired by these results, we explore whether repeated sampling can similarly benefit *conversational recommendation systems*, where the goals differ in two critical ways: (1) there is no ground-truth verifier that can score outputs as "correct" or "incorrect," and (2) the quality of a recommendation depends heavily on alignment with user intent, which is often implicit, ambiguous, or evolving throughout the dialogue.

In this work, we investigate the application of repeated sampling in *multi-turn recommendation dialogues*, where an LLM generates multiple candidate next-turn responses given the conversational context. We then evaluate the potential gains in recommendation quality by using lightweight selection strategies such as user preference heuristics, semantic similarity, or majority content voting.

Our contributions are as follows:

- We show that repeated sampling can significantly improve recommendation relevance in dialogue systems without requiring additional model training or fine-tuning.
- We compare different selection strategies for choosing among samples in the absence of a gold verifier and highlight the conditions under which each is most effective.
- We demonstrate that sampling from smaller or cheaper models multiple times can outperform stronger models in single-attempt settings, mirroring findings in coding tasks but applied to subjective human-centered evaluations.

While our work builds upon the foundational approach introduced in *Large Language Monkeys* [1], we focus on a fundamentally different domain where recommendation quality cannot be measured purely through correctness, and verification is a softer, human-aligned challenge.

IIC3633 Conference'25, July 2025, Santiago, Chile

2 State of the Art

The rise of large language models (LLMs) has prompted a wave of research into their integration with recommender systems (RS), especially in conversational settings. A growing body of survey work provides overviews of this intersection from various angles. For example, [7] and [6] discuss how LLMs can enhance RS pipelines including user modeling, item understanding, and interaction design. [9] and extend this to explainability and personalization, while [13] focuses on multimodal recommendation. A broader conceptual overview is offered by [3].

While these surveys highlight emerging architectures and applications, they do not systematically investigate the behavior of LLMs under repeated decoding, nor the implications of sampling variance in generation-based CRS. Our work builds directly on *Large Language Monkeys* [1], which introduced repeated sampling as a method to boost performance in general-purpose LLM tasks. However, [1] did not evaluate this technique in the context of recommendation, nor did it examine conversational settings, personalization effects, or recommendation-specific metrics.

On the dataset side, LLM-REDIAL [11] provides a large-scale CRS benchmark based on simulated user conversations, and is used in our experiments. From an evaluation standpoint, prior work has explored different aspects of uncertainty in LLMs [8, 10, 12], as well as limitations of standard recommendation metrics such as popularity bias [2], tie handling in rankings [4], and fairness or exposure imbalances [5, 14].

Our work aims to bridge these directions by applying repeated sampling to CRS tasks, and by providing a metric-centered analysis across diversity, uncertainty, and user alignment. To our knowledge, this is the first study to do so in a systematic and reproducible fashion.

3 Dataset

We base our experiments on the REDIAL dataset [11], a benchmark corpus for training and evaluating conversational movie recommender systems. REDIAL consists of over 10,000 dialogues between two crowdworkers: a recommendation seeker and a recommender. Each dialogue revolves around the seeker's request for movie suggestions tailored to their tastes, often including explicit and implicit feedback throughout the conversation.

The dataset is particularly well-suited to our study due to its multi-turn structure and focus on subjective, user-aligned recommendation quality. Each dialogue includes metadata such as movie mentions (linked to IMDb), utterance-level speaker roles, and optional sentiment labels indicating whether a movie was liked, disliked, or seen. This structure allows us to test recommendation generation in a realistic conversational context, incorporating both explicit user preferences and contextual cues from prior turns.

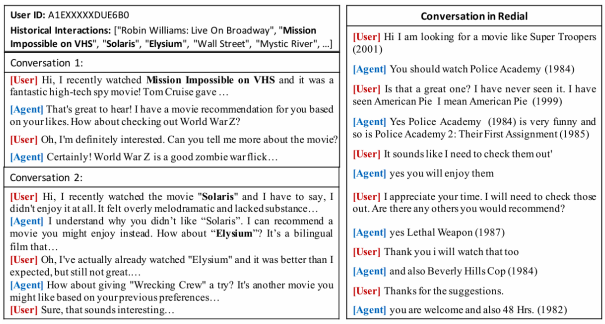


Figure 1: Examples of dialogues in LLM-REDIAL and REDIAL. Reproduced from [11] under the CC-BY 4.0 license.

4 Methodology

We evaluated the effectiveness of repeated sampling in conversational recommendation by testing multiple models under controlled configurations.

We focused our analysis on two families of language models:

- **TinyLlama 1.1B**: A compact, open-weight model based on the LLaMA architecture, trained to be efficient and capable on resource-constrained hardware. Despite its small size (1.1 billion parameters), TinyLlama has been shown to perform competitively on various natural language understanding tasks.
- **Gemma 2B Instruct**: Released by Google DeepMind, Gemma is another lightweight, instruction-tuned open-source model optimized for alignment and helpfulness in multi-turn dialogues. With 2 billion parameters, it strikes a balance between scale and usability.

Both TinyLlama and Gemma are freely available under open licenses, making them attractive for academic and real-world use cases without incurring commercial costs. In contrast, the GPT-4.1-mini and GPT-4.1-nano models used as baselines are proprietary offerings from OpenAI. These commercial models are not open-weight and can only be accessed through paid APIs or licensed platforms, which restricts reproducibility and limits transparency.

Our primary focus was to generate $k = 20$ responses per test conversation. In each response, the models produced a list of 10 movie recommendations conditioned on the multi-turn dialogue context provided.

Three distinct inference configurations were explored:

- **Zero-Shot**: No examples or additional context beyond the current dialogue were given.
- **Few-Shot**: Each prompt included four example dialogue-recommendation pairs as in-context demonstrations.
- **Fine-Tuned**: We fine-tuned TinyLlama on 50% of the LLM-REDIAL training set and Gemma on 100%, using Low-Rank Adaptation (LoRA) to reduce memory footprint and training cost.

As baselines, we used GPT-4.1, GPT-4.1-nano and GPT-4.1-mini. These were evaluated using single-response inference ($k = 1$), under both Zero-Shot and Few-Shot settings.

Each generated list was evaluated against ground truth using the following standard metrics: Recall@1, Recall@5, and Recall@10 to assess whether the target movie appeared within the top 1, 5, or 10 positions, and NDCG@5, and NDCG@10 to capture how highly ranked the ground truth recommendation was.

Given that only a single ground-truth movie is provided per conversation, Recall serves as a binary indicator of presence, while NDCG accounts for the relative rank of the correct recommendation.

To analyze the effect of sampling, we also included Uncertainty, Diversity and Novelty metrics. All evaluations were conducted using a held-out test set of 100 dialogues from LLM-REDIAL, focusing exclusively on configurations with user interaction history, as preliminary tests showed significant performance degradation in its absence. This choice enabled a more realistic assessment of personalization and alignment with user preferences. The previously mentioned metrics are defined in the sections that follow.

4.1 Uncertainty

Uncertainty has been a topic of investigation for many academic papers ([8, 10, 12]). Kuhn et al. define uncertainty as the predictive entropy of the output distribution. This in turn measures the information you have about the output, given the input. Over the years, there have been many attempts at quantifying the uncertainty of LLMs ([8, 10, 12]). For our purposes we found that the method proposed by Lin et al. was a natural fit for our prompt sampling approach. In Lin et al.'s investigation they found that uncertainty can be quantified by taking the following steps:

1. For a given input x, generate k response samples $s_1, ..., s_k$.
2. Calculate the pairwise similarity scores $a(s_{j_1}, s_{j_2})$ for these k responses.
3. Compute an uncertainty estimate $U(x)$ using the similarity values.

To measure response similarities we use Rank-Biased Overlap (RBO) [4] as opposed to Jaccard similarity because as explained by Lin et al., Jaccard similarity does not consider order within a list while RBO is a top-weighted similarity measure for ranking lists. Considering all this, we compute uncertainty for a given input x as the average pairwise distance $D_{i,j}$ for all N pairs:

$$U(x) = \frac{1}{N} \sum_{i<j\leq k} D_{i,j} \tag{1}$$

or in terms of similarity:

$$U(x) = 1 - \frac{1}{N} \sum_{i<j\leq k} S_{i,j} \tag{2}$$

Where $S_{i,j}$ is the pairwise similarity for the k sample responses for input x for all N pairs using RBO.

4.2 Diversity

In this research we use the diversity metric as a measure of popularity bias within all recommended movies. It tries to measure if there are movies which are globally recommended more than others. Many authors have tackled the issue of popularity bias in recommender systems ([2, 5, 14]). One common way to measure this popularity bias is by using the Gini index as used by Braun et al. and Sun et al.. The Gini coefficient was originally developed to measure income inequality within a population [2], however as previously mentioned it has been recently used to measure popularity bias in the context of recommendation systems. The Gini coefficient is computed as:

$$G = \frac{\sum_{i=1}^{n}(2i - n - 1)\phi_i}{n \sum_{i=1}^{n} \phi_i} \tag{3}$$

Where ϕ_i is the popularity score of item i. For our project, we define the popularity score as the *global* ratio of number of lists which contain item i by the total number of recommendation lists, i.e.:

$$\phi_i = \frac{N_i}{N_{total}} \tag{4}$$

Where i is an item, N_i is the number of lists which contain item i, and N_{total} is the total number of lists. We then define our Diversity score as the complement of the Gini index:

$$Div = 1 - G \tag{5}$$

Where scores closer to 1 mean a higher diversity of recommended items while scores closer to 0 mean that the recommendations are dominated by only a few items.

4.3 Novelty

Novelty is a metric which is closely related to diversity, especially within the context of popularity-based novelty and popularity-based diversity. Authors such as Vargas and Castells use the inverse log of an item popularity as a measure of popularity based novelty, where high novelty values correspond to movies which have seldom been recommended by the model, and low novelty values correspond to movies which the recommendation system has disproportionately recommended to users. The novelty score we use is the average of the global novelty scores of the movies within a recommended item list for input x:

$$Nov(x) = -\frac{1}{k} \sum_{i=1}^{k} \log_2(f_i) \tag{6}$$

Where x is a given input, i is a given item, k is the number of recommendation lists per input and f_i is the frequency of a given item i, defined as the ratio of the number of times item i has been recommended by the total number of recommended movies, considering all times a given item i has been recommended.

5 Evaluation Criteria and Examples

To evaluate the quality of the recommendations generated by the models, we compared the top-k list produced by the model for each conversation against the ground truth movie referenced by the user at the end of the dialogue.

We used the following criteria:

- **Direct Match (Hit Rate):** A recommendation list is considered a hit if it contains the exact ground truth movie anywhere among the top-k items. This strict criterion measures the model's ability to reproduce the user's referenced movie directly.
- **Rank-Biased Overlap (RBO):** We use the RBO metric to measure the similarity between the ranked list generated by the model and a reference list containing the ground truth movie at the top. RBO accounts for the ranked positions of matches, giving higher weight to agreements at top ranks, and thus provides a graded measure of similarity beyond strict hits. This metric also helps us analyze uncertainty by quantifying partial overlaps in recommendation rankings.

Example 1: Good Recommendation List

Ground truth movie: *Notting Hill VHS*
Model-generated list:

```
[ "Notting Hill VHS", "The Quiet Man", "Friends
- The Complete Series Collection DVD", "Captain
America : First Avenger", "Shogun", "The Lion
King 1 1/2", "Erin Brockovich VHS", "Johnny
Tremain VHS", "Willy Wonka & the Chocolate Factory
VHS" ]
```

In this case, the ground truth movie appears at the very top of the list, resulting in a direct hit and a maximal RBO score close to 1. This indicates high confidence and accuracy in the model's recommendation.

Example 2: Poor Recommendation List

Ground truth movie: *Notting Hill VHS*
Model-generated list:

```
[ "The Godfather I", "The Godfather II", ...,
"The Godfather X" ]
```

Here, the model repeatedly recommends variations of "The Godfather" sequels, many of which do not exist. This behavior likely stems from the model falling back on popular, well-known titles when uncertain, producing repetitive and low-diversity outputs. As a result, the ground truth movie is absent, leading to zero direct hits and low RBO scores.

We addressed this problem primarily through prompt engineering, guiding the model to avoid repetitive suggestions and promote more diverse and relevant recommendations. This significantly reduced cases of repeated or overly similar movie titles in generated lists.

5.1 Application

These evaluation criteria were applied uniformly across all test samples. Hit rate was calculated as the fraction of generated lists containing the ground truth movie. RBO was computed using a weight parameter $p = 0.9$ to emphasize top-ranked items, enabling a nuanced assessment of list similarity and model uncertainty. Together, these metrics provided a quantitative basis for comparing different generation strategies, model variants, and reranking methods.

5.2 Prompt Analysis

To assess the sensitivity of our models to prompt phrasing, we conducted a controlled prompt analysis using two distinct input formulations. Prompt 1 was written in English and specified detailed output constraints in a rule-based style, closely resembling instruction-tuned input for LLMs:

```
You are a Movie Recommendation System.
Generate a numbered list of 10 Movies.
RULES:
a) DO NOT write dialogs, explanations nor additional
text or information.
b) DO NOT recommend movies already mentioned
in the conversation.
c) You MUST recommend 10 movies, nothing more,
nothing less.
d) The movies MUST be numbered from 1 to 10,
with one movie name per line.
Failure to follow the rules will result in incorrect
output and be discarded by the system.
```

Prompt 2, by contrast, was a simpler and more natural prompt written in Spanish, mimicking a colloquial user query:

```
Me das una lista de 10 películas que recomiendes?
Solamente dame la lista
de las 10 películas enumeradas del 1 al 10, con
sus nombres en inglés, y
sin repetir la misma película en la lista. No
digas nada más.
```

We tested both prompts across the Gemma and TinyLlama models to measure their robustness to input variation. The generated outputs were evaluated solely on format correctness, meaning whether the output adhered to the expected structure of a list with exactly 10 distinct, numbered movie names in English and no additional text.

The results revealed substantial variation between models. For **Gemma**, Prompt 2 yielded malformed or non-conforming outputs in **70% of the cases**, while Prompt 1 significantly improved adherence to the format, with only **4% of outputs** failing to meet the criteria. Based on this analysis, we decided to use Prompt 1 exclusively for all subsequent Gemma evaluations.

For **TinyLlama**, the model exhibited higher prompt robustness: Prompt 1 achieved **0%** malformed outputs, while Prompt 2 had a slightly higher failure rate of **1%**. Despite both prompts being viable for TinyLlama, we adopted Prompt 1 to maintain consistency across models and scenarios.

These findings suggest that TinyLlama exhibits greater resilience to prompt phrasing variation, while Gemma benefits more from highly structured, rule-based prompts. This insight highlights the importance of prompt engineering when working with lightweight LLMs, especially in tasks requiring strict output formatting.

6 Results

Table 1: Obtained metrics evaluating the models with 100 test items.

Model	#samp(k)	R@1	R@5	R@10	N@5	N@10
GEMMA						
Zero-Shot	k = 1	0.0000	0.0000	0.0106	0.0000	0.0042
	k = 10	0.0213	0.0426	0.0426	0.0316	0.0302
	k = 20	0.0213	0.0426	0.0426	0.0316	0.0302
Few-Shot	k = 1	0.0105	0.0211	0.0316	0.0157	0.0194
	k = 10	0.0105	0.0316	0.0316	0.0208	0.0198
	k = 20	0.0105	0.0316	0.0316	0.0208	0.0198
Fine-Tuned	k = 1	0.0000	0.0100	0.0200	0.0052	0.0091
	k = 10	0.0300	0.0500	0.0500	0.0401	0.0364
	k = 20	**0.0300**	**0.0500**	**0.0500**	**0.0401**	**0.0364**
TinyLlama						
Zero-Shot	k = 1	0.0000	0.0700	0.0700	0.0340	0.0290
	k = 10	0.0200	0.1400	0.1500	0.0820	0.0820
	k = 20	0.0200	0.1500	0.1700	0.0870	0.0910
Few-Shot	k = 1	0.0000	0.0000	0.0000	0.0000	0.0000
	k = 10	0.0200	0.0200	0.0400	0.0200	0.0280
	k = 20	0.0200	0.0300	0.0400	0.0250	0.0280
Fine-Tuned	k = 1	0.1200	0.1300	0.1300	0.1250	0.1240
	k = 10	0.1500	0.1500	0.1500	0.1500	0.1500
	k = 20	**0.1600**	**0.1900**	**0.1900**	**0.1750**	**0.1730**
GPT-4.1 nano						
Zero-Shot	k = 1	**0.0100**	**0.0100**	0.0300	**0.0100**	**0.0180**
Few-Shot	k = 1	0.0000	0.0100	**0.0400**	0.0050	0.0160
GPT-4.1 mini						
Zero-Shot	k = 1	0.0100	0.0200	0.0200	0.0150	0.0140
Few-Shot	k = 1	**0.0200**	**0.0200**	**0.0200**	**0.0200**	**0.0200**
GPT-4.1						
Zero-Shot	k = 1	**0.0200**	**0.0300**	**0.0300**	**0.0250**	**0.0240**
Few-Shot	k = 1	0.0200	0.0200	0.0300	0.0200	0.0240

Table 2: Comparison of uncertainty, diversity, and novelty metrics for Gemma and TinyLlama across different settings.

	Gemma			TinyLlama		
Setting	**Uncert.**	**Diver.**	**Novel.**	**Uncert.**	**Diver.**	**Novel.**
Zero-Shot	0.648	0.542	**0.352**	**0.611**	0.507	**0.694**
Few-Shot	0.659	**0.555**	0.321	0.814	**0.595**	0.587
Fine-Tuned	**0.521**	0.433	0.297	0.837	0.547	0.543

Our main results are shown in Table 1. As we can see, the experimental results reveal how using a repeated sampling approach affects the performance of different language models in recommendation tasks. Our results show a marked improvement in performance across all models with the usage of repeated sampling strategies; however, the magnitude of performance improvement varies considerably across different model architectures.

TinyLlama demonstrates the most pronounced sensitivity to sampling among all evaluated models. The performance improvement from single-sample to multi-sample generation is substantial, with R@1 scores increasing from 0.00 to 0.02 and R@10 scores improving from 0.07 to 0.17 in a zero-shot setting. In a fine-tuned setting, TinyLlama performs the best across all metrics even with single-sample generation, but performance from single-sample to multi-sample also improves substantially, with R@1 increasing from 0.12 to 0.16 and R@10 improving from 0.13 to 0.19.

In contrast, the GPT-4.1 family of models shows relatively stable performance among themselves. However, when comparing these much larger commercial models, especially GPT-4.1, to the performance of a small model such as TinyLlama, we observe that while TinyLlama performs worse in a single-sample setting for R@1 and similarly for other metrics, using a repeated sampling approach allows TinyLlama to perform comparably to the GPT-4.1 family at worst, and significantly outperform them across all metrics at best.

Gemma, on the other hand, shows mixed behavior with only moderate improvements when increasing sampling in certain configurations. We speculate that this difference in performance between Gemma and TinyLlama is due to differences in their model architectures and the type of data used to train them.

As mentioned previously we also evaluated these models with Uncertainty, Diversity and Novelty metrics. Those results are shown in Table 2. In terms of Uncertainty, Gemma shows a consistently more confident behavior than TinyLlama in both Few-Shot and Fine-Tuned settings. TinyLlama in turn shows a remarkably poor performance in terms of its Uncertainty score in the previously mentioned settings. This ties in to the Diversity scores shown in the table, where we can see a consistent increase in Diversity with the increase of Uncertainty, which could explain why TinyLlama performs so much better with sampling: the more diverse the samples, the more opportunity there is to generate a better recommendation list.

Interestingly, Gemma shows remarkably lower Novelty scores than TinyLlama, suggesting that it's recommending some movies disproportionately more than others. TinyLlama on the other hand shows more balanced scores in Few-Shot and Fine-Tuned settings, with the Zero-Shot setting showing a high Novelty score which suggests that the movies it's recommending are not popular within the other lists.

Correlating Uncertainty and performance metrics we can see in Figures 2 and 3 that especially in a Fine-Tuning setting there's a clear correlation between Uncertainty and model performance with a Pearson Correlation score of -0.93 for both Recall@1 and NDCG@10 with response sampling.

Figure 2: Correlation between average cumulative Uncertainty and average cumulative Recall@1 for TinyLlama in a Fine-Tuned setting with sampling.

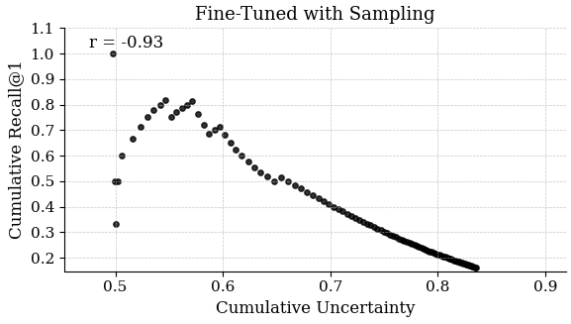
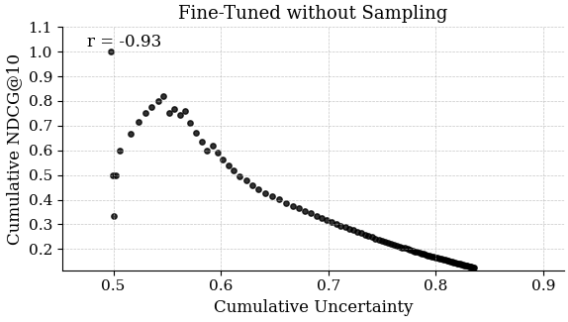


Figure 3: Correlation between average cumulative Uncertainty and average cumulative NDCG@10 for TinyLlama in a Fine-Tuned setting with sampling.



7 Conclusions

In this work, we explored how repeated sampling strategies can enhance the performance of lightweight large language models (LLMs) in conversational recommendation tasks. Motivated by the success of sampling techniques in other domains, we applied this paradigm to the generation of movie recommendations in dialogue settings using the LLM-REDIAL dataset.

Our findings demonstrate that repeated sampling leads to consistent improvements across standard recommendation metrics such as Recall and NDCG. Notably, TinyLlama despite being a compact model, achieved competitive or superior performance compared to larger commercial models like GPT-4.1-mini and nano when multiple samples were used. This supports the notion that sampling can serve as a cost-effective alternative to scaling model size in low-resource or constrained environments.

In addition, our analysis across Uncertainty, Diversity, and Novelty revealed that models with higher output variability benefit the most from repeated sampling. TinyLlama, in particular, showed strong performance gains due to its greater output diversity. In contrast, Gemma displayed more deterministic behavior and benefitted less from sampling, highlighting differences in architectural and training dynamics.

Our prompt analysis further emphasized the sensitivity of lightweight models to prompt phrasing. Gemma, for instance, required more structured and rule-based prompts to produce consistent outputs, whereas TinyLlama was more robust to naturalistic or loosely structured inputs. This underscores the importance of prompt engineering as a complementary factor to model selection.

Overall, our results suggest that repeated sampling combined with lightweight model architectures and careful prompt design can unlock significant potential for building scalable, accessible, and high-quality conversational recommender systems without the need for expensive infrastructure or proprietary models.

Future work may explore more sophisticated reranking strategies, user-aligned reward functions, and real-time deployment scenarios to further validate the practical viability of sampling-enhanced conversational recommenders.

References

[1] arXiv 2024. Large Language Monkeys: Scaling Inference Compute with Repeated Sampling. *arXiv preprint arXiv:2407.21787* abs/2407.21787, 1 (2024), 12 pages.

[2] Valentijn Braun, Debarati Bhaumik, and Diptish Dey. 2023. Metrics for popularity bias in dynamic recommender systems. arXiv:2310.08455 [cs.CY] https://arxiv.org/abs/2310.08455

[3] Yixin Cao, Yongfeng Zhang, et al. 2023. Recommender Systems in the Era of Large Language Models (LLMs). *arXiv preprint arXiv:2307.02046* (2023). https://arxiv.org/pdf/2307.02046

[4] Matteo Corsi and Julián Urbano. 2024. The Treatment of Ties in Rank-Biased Overlap. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2024).* ACM, 251–260. https://doi.org/10.1145/3626772.3657700

[5] Yashar Deldjoo, Dietmar Jannach, Alejandro Bellogin, Alessandro Difonzo, and Dario Zanzonelli. 2023. Fairness in recommender systems: research landscape and future directions. *User Modeling and User-Adapted Interaction* 34, 1 (April 2023), 59–108. https://doi.org/10.1007/s11257-023-09364-z

[6] Haoran Gao, Meng Liu, Jiahua Liu, and Yongfeng Zhang. 2024. Towards Next-Generation LLM-based Recommender Systems: A Survey and Beyond. *arXiv preprint arXiv:2410.19744* (2024). https://arxiv.org/pdf/2410.19744

[7] Haoran Gao and Yongfeng Zhang. 2023. How Can Recommender Systems Benefit from Large Language Models: A Survey. *arXiv preprint arXiv:2306.05817* (2023). https://arxiv.org/pdf/2306.05817

[8] Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2024. Decomposing Uncertainty for Large Language Models through Input Clarification Ensembling. arXiv:2311.08718 [cs.CL] https://arxiv.org/abs/2311.08718

[9] Chen Hu, Shaohua Wang, Jiajun Zhang, Huifeng Wang, and Xiaoming Li. 2023. A Survey on Large Language Models for Personalized and Explainable Recommendations. *arXiv preprint arXiv:2311.12338* (2023). https://arxiv.org/pdf/2311.12338

[10] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic Uncertainty: Linguistic Invariances for Uncertainty Estimation in Natural Language Generation. arXiv:2302.09664 [cs.CL] https://arxiv.org/abs/2302.09664

[11] Tingting Liang, Chenxin Jin, Lingzhi Wang, Wenqi Fan, Congying Xia, Kai Chen, and Yuyu Yin. 2024. LLM-REDIAL: A Large-Scale Dataset for Conversational Recommender Systems Created from User Behaviors with LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024.* Association for Computational Linguistics, Bangkok, Thailand, 8380–8396. https://aclanthology.org/2024.findings-acl.529

[12] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with Confidence: Uncertainty Quantification for Black-box Large Language Models. arXiv:2305.19187 [cs.CL] https://arxiv.org/abs/2305.19187

[13] Jiashuo Qiu, Xuhui Li, Jianzong Wang, Yuanhang Liu, Xin Jin, Yang Yang, Min Zhang, Wayne Xin Zhao, and Hongbo Wang. 2025. A Survey on Large Language Models in Multimodal Recommender Systems. *arXiv preprint arXiv:2505.09777* (2025). https://arxiv.org/pdf/2505.09777

[14] Wenlong Sun, Sami Khenissi, Olfa Nasraoui, and Patrick Shafto. 2019. Debiasing the Human-Recommender System Feedback Loop in Collaborative Filtering. In *Companion Proceedings of The 2019 World Wide Web Conference (WWW '19).* ACM, 645–651. https://doi.org/10.1145/3308560.3317303

[15] Saúl Vargas and Pablo Castells. 2011. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems* (Chicago, Illinois, USA) *(RecSys '11).* Association for Computing Machinery, New York, NY, USA, 109–116. https://doi.org/10.1145/2043932.2043955