MEJORANDO RECOMENDACIONES CONVERSACIONALES CON MUESTREO REPETIDO EN LLMS: EXTENDIENDO EL PARADIGMA DE LOS LLMONKEYS

Joaquín Peralta | Nicolás Salazar | Tomás Trincado

PROBLEMA

Los LLMs han mejorado significativamente las recomendaciones conversacionales, pero la mayoría de los sistemas actuales se basan en una única inferencia por prompt.

Estudios recientes muestran que generar múltiples respuestas por consulta (muestreo repetido) permite a modelos pequeños superar a modelos grandes en tareas complejas.

Este trabajo evalúa si esa estrategia también mejora la calidad de recomendaciones en contexto conversacional, usando modelos ligeros y comparando su rendimiento con un modelo comercial de mayor capacidad.

OBJETIVOS

Evaluar si el muestreo repetido mejora la calidad de las recomendaciones generadas por LLMs en diálogos multi-turno.

Comparar el rendimiento de modelos pequeños (TinyLlama 1.1B, Gemma 2B) con modelos comerciales (GPT-4.1-nano, GPT-4.1-mini y GPT4.1) sin muestreo.

Explorar estrategias simples de selección entre muestras generadas, considerando alineamiento semántico y preferencias del usuario.

DATASET: LLM-REDIAL

Es el mayor corpus disponible de diálogos multi-turno generados a partir de comportamientos reales de usuarios.

A diferencia de otros datasets, LLM-REDIAL garantiza consistencia semántica entre los diálogos y las interacciones históricas del usuario.

(a) Dialogue Template

[User] [Greeting with [USER_HIS_LIKES_1] and [USER_HIS_LIKES_REVIEW_1]]
[Agent] [Responds with [OTHER_HIS_LIKES_REVIEW_1]]
[User] [Ask for the recommendation]
[Agent] [Recommend [USER_MIGHT_LIKE]]
[User] [Express interest][Ask more for the movie detail]
[Agent] [Responds in detail with [OTHER_HIS_MIGHT_LIKES_REVIEW]]
[User] [Accept recommend with the reason]
[Agent] [End the conversation reasonably]

MODELOS

Se evaluaron 5 modelos de lenguaje con capacidad conversacional:

GPT4.1, GPT-4.1-nano y GPT-4.1-mini (baseline)
Modelos comerciales cerrados, de arquitectura desconocida, utilizados como línea base.

TinyLlama 1.1B
Modelo open-source de arquitectura LLaMA reducido (1.1B parámetros), optimizado para eficiencia y despliegue local.

Gemma 2B (google/gemma-2b-it)
Modelo de código abierto desarrollado por Google, con 2B parámetros, entrenado con enfoque instructivo.

PROCEDIMIENTO

Se evaluaron dos modelos ligeros (TinyLlama y Gemma) generando 20 muestras por prompt (k = 20) para cada conversación del set de testeo. En cada muestra, el modelo recomendó una lista de 10 películas basada en el contexto conversacional entregado.

Se consideraron tres configuraciones:

- **Zero-Shot**: sin ejemplos previos ni contexto adicional.
- **Few-Shot**: se incluyeron 4 ejemplos de recomendaciones exitosas antes de cada prompt.
- **Fine-Tuning:**
  - TinyLlama fue entrenado con el 50 % del set de entrenamiento.
  - Gemma fue entrenado con el 100 %.

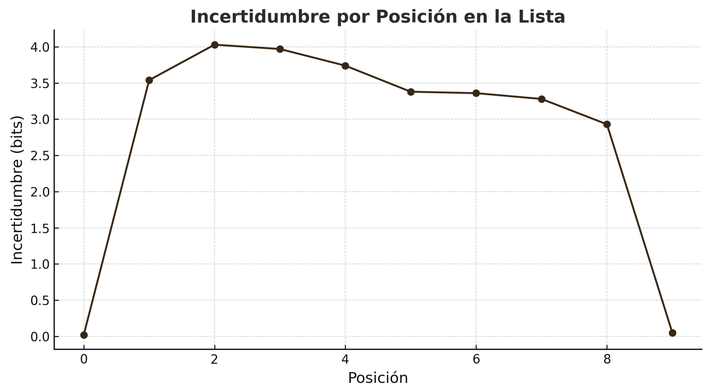
Como líneas base (k = 1), se utilizaron los modelos comerciales GPT-4.1-mini (8B) y GPT-4.1-nano, evaluados en los escenarios Zero-Shot y Few-Shot sin muestreo repetido.

Cada modelo se evaluó con métricas de Recall@1, Recall@5 y Recall@10, y NDCG@1, NDCG@5 y NDCG@10. Al haber una única película correcta (ground truth) por conversación evaluada, Recall nos dice si dicha película aparece o no dentro de las primeras 1, 5 y 10 películas mencionadas, y NDCG evalúa la posición en la que se encuentra dicha película. Además, se calcularon métricas de Incertidumbre entre las 20 listas generadas por cada conversación.

RESULTADOS

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

El mejor desempeño lo obtiene TinyLlama 1.1B con Fine-Tuning y muestreo repetido, alcanzando un Recall@1 de 0.1600 y NDCG@10 de 0.1730. Esta configuración supera ampliamente al baseline comercial GPT-4.1, que alcanza un máximo de 0.0200 en Recall@1.
En general, el Fine-Tuning es la estrategia que más mejora el rendimiento, seguido por el muestreo repetido, que también aporta mejoras incluso sin entrenamiento adicional.
En configuración Zero-Shot + muestreo, los modelos ligeros alcanzan un rendimiento cercano al baseline comercial, mostrando el potencial de este enfoque en escenarios sin Fine-Tuning.



TinyLlama con Fine-Tuning presenta incertidumbre casi nula en la primera posición, lo que indica una alta confianza en su recomendación inicial. Sin embargo, la incertidumbre aumenta rápidamente en las siguientes posiciones, lo que sugiere que las recomendaciones posteriores son menos consistentes entre muestras y potencialmente menos alineadas con las preferencias del usuario.

CONCLUSIONES: INTENTAR HASTA CONQUISTAR

Observamos que el muestreo repetido (generar múltiples respuestas por prompt) puede permitir que modelos ligeros y de código abierto superen en rendimiento a modelos comerciales más grandes en tareas de recomendación conversacional.

Esto se evidencia especialmente en el caso de TinyLlama 1.1B, que tras un proceso de Fine-Tuning y muestreo repetido (20 muestras), logró un Recall@1 de 0.16 y un NDCG@10 de 0.173, superando a las versiones comerciales evaluadas (GPT-4.1-nano, GPT-4.1-mini y GPT-4.1) en sus configuraciones Zero-Shot y Few-Shot sin muestreo.

REFERENCIAS

- Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Re, C., & Mirhoseini, A. (2024).
- Large Language Monkeys: Scaling Inference Compute with Repeated Sampling. arXiv preprint arXiv:2407.21787. Recuperado de https://arxiv.org/pdf/2407.21787v1
- Liang, T., Jin, C., Wang, L., Fan, W., Xia, C., Chen, K., & Yin, Y. (2024). LLM-REDIAL: A Large-Scale Dataset for Conversational Recommender Systems Created from User Behaviors with LLMs. Findings of the Association for Computational Linguistics: ACL 2024, 8926–8939. Recuperado de https://aclanthology.org/2024.findings-acl.529.pdf
- Kong, Y., Liao, Q. V., Krishnamurthy, J., Li, Z., Li, X. L., Zhang, Y. (2023). A Large Language
- Model for Conversational Recommendation (arXiv preprint arXiv:2311.08718). Recuperado de: https://arxiv.org/abs/2311.08718
- Peralta, J., Salazar, N., & Trincado, T. (2025). monkey-recsys [Repositorio de GitHub]. Recuperado de https://github.com/roahoki/monkey-recsys