Text Is All You Need: Learning Language Representations for Sequential Recommendation

Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley.
Presentado por: Javiera Belén López Massaro, Javiera Paz Azócar Oliva, Pablo Poblete Arraé, Gabriel Acevedo

KDD 2023

Contexto

Sistemas de recomendación secuencial

- Sistemas que buscan modelar el comportamiento dinámico del usuario a partir de interacciones históricas para sugerir ítems de interés.
- Su objetivo es recomendar ítems potenciales que interesen a los usuarios.
- Tienen la capacidad de capturar tanto las preferencias a corto como a largo plazo de los usuarios. Por ello, son ampliamente utilizados en diversos escenarios de recomendación.

Problema

- Los modelos secuenciales de recomendación están basados en el ID del item, por lo que al añadir nuevos items nos vemos enfrentados al cold-start
- Un segundo problema que ocurre con los sistemas secuenciales actuales [Cadenas de Markov, RNN], es la poca flexibilidad, al tener una baja capacidad de transferir el conocimiento a otros contextos
- Representaciones textuales como Bert no capturan preferencias tan finas, al fijase más en lo semantico

Contribución

- Formulan los ítems como pares de atributos clave-valor para la recomendación secuencial sin ID y proponen una novedosa estructura Transformer bidireccional para codificar secuencias de pares clave-valor.
- Diseñan el marco de aprendizaje que ayuda al modelo a conocer las preferencias de los usuarios y, posteriormente, a recomendar ítems basándose en representaciones lingüísticas y a transferir el conocimiento a diferentes dominios de recomendación e ítems de inicio en frío.
- Se realizan experimentos exhaustivos para demostrar la eficacia del método.

Marco Teórico de Recformer

- **Modelo Secuencial**: predice el siguiente ítem basado en el historial del usuario.
- **Representación Textual**: usa descripciones textuales estructuradas (título, marca, categoría) en lugar de IDs.
- **Codificación Semántica**: emplea encoders preentrenados como BERT para generar embeddings significativos.
- **Modelado con Transformers**: utiliza arquitectura Longformer para procesar secuencias largas con atención eficiente.
- **Predicción por Similitud**: compara intención del usuario con todos los ítems candidatos usando producto punto.
- **Generalización y Transferencia**: diseño modular facilita adaptabilidad a dominios nuevos y problemas de cold-start.

Metodología - Formulación del problema

- En una secuencia de interacciones $s = \{i_1, i_2, ..., i_n\}$. Cada ítem i tiene un diccionario de atributos D_i con pares clave-valor (ej., {Title: "MacBook Air Laptop M1 Chip", Brand: "Apple", Color: "Gold"}).

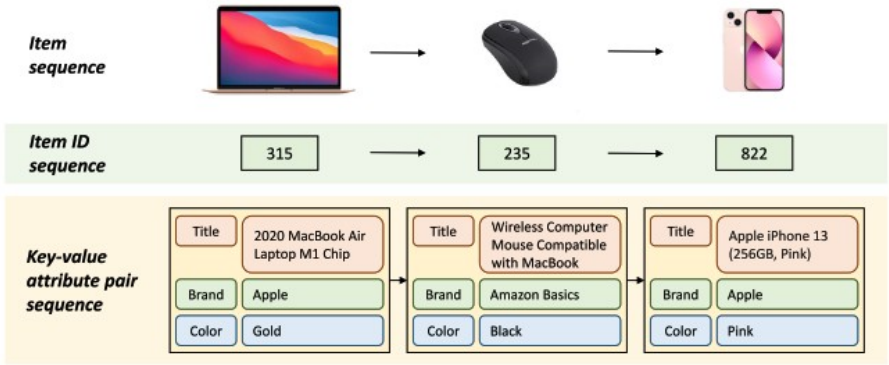

Figura: Comparación de Input

Metodología

Construcción de la entrada del modelo:

- Los pares clave-valor se "aplanan" en una "sentencia" del ítem $Ti = \{k_1, v_1, k_2, v_2, ..., k_m, v_m\}$. Una secuencia de interacción de usuario se invierte $(i_n, i_{n-1}, ..., i_1)$ y se añade un token especial $[CLS]$ al principio, formando la entrada $X = \{[CLS], Tn, Tn-1, ..., T1\}$.

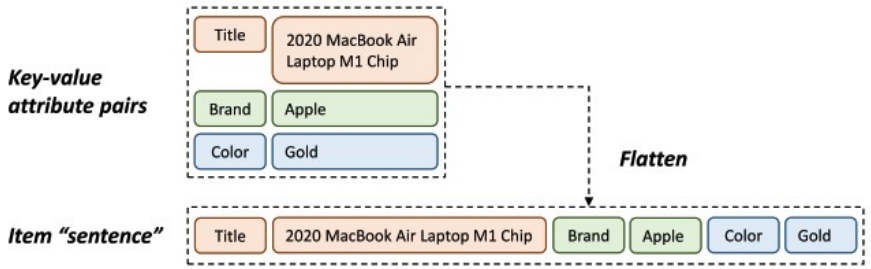

Figura: Item sentences entrada del modelo

Metodología - Arquitectura de Recformer

- Un Transformer bidireccional de múltiples capas basado en la estructura de Longformer. Utiliza atención de ventana local para secuencias largas, con el token [*CLS*] teniendo atención global

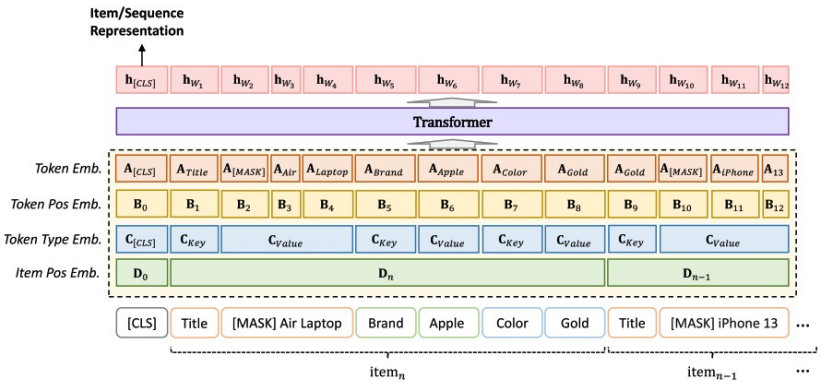

Figura: Modelo Recformer

Arquitectura de Recformer

- Basado en Longformer: atención eficiente para secuencias largas.
- 4 embeddings:
  1. Token embedding (significado básico de la palabra)
  2. Posición del token (palabra dentro de la frase del Item)
  3. Tipo del token (palabra es clave o valor, [CLS])
  4. Posición del item (indica a que item pertenece la palabra)
- Representaciones derivadas del token [CLS] (entendimiento del modelo sobre las preferencias del usuario)

Objetivos de Entrenamiento

- El procedimiento de entrenamiento incluye dos etapas:
  1. Preentrenamiento: se aprenden representaciones generales usando datos a gran escala.
  2. Fine-tuning: se adapta el modelo a un dominio específico.
- En ambas etapas se utiliza como objetivo principal la predicción del siguiente ítem.

Preentrenamiento

- Se utilizan dos objetivos principales:
  - **MLM (Masked Language Modeling):** ayuda a preservar la semántica del texto del ítem.
  - **Aprendizaje Contrastivo entre Ítems:** maximiza la similitud (coseno) entre el vector de contexto y el ítem real siguiente, diferenciándolo de ítems negativos.

Fine-tuning en Dos Etapas

- **Etapa 1:** se congela el codificador textual. Solo se entrena el codificador secuencial y la capa de predicción.
  - Las representaciones de ítems se recalculan en cada época.
- **Etapa 2:** se ajusta todo el modelo incluyendo el codificador textual para una mejor adaptación al dominio destino.

Algoritmo 1: Fine-tuning en Dos Etapas

1: Inicializar el modelo Recformer con pesos preentrenados.
2: Congelar encoder de texto.
3: **for** cada batch de entrenamiento (Etapa 1) **do**
4: Propagación hacia adelante con el Transformer.
5: Calcular pérdida y retropropagación.
6: Actualizar parámetros del Transformer.
7: **end for**
8: Descongelar encoder de texto.
9: **for** cada batch de entrenamiento (Etapa 2) **do**
10: Propagación hacia adelante completa.
11: Calcular pérdida total y retropropagación.
12: Actualizar todos los parámetros.
13: **end for**

Resumen Paso a Paso del Algoritmo

1. Congelar encoder textual = evitar modificar representaciones base.
2. Entrenar solo el codificador secuencial permite adaptación a dominio sin afectar semántica textual.
3. Luego, desbloquear encoder textual permite ajuste fino conjunto.
4. Este procedimiento mejora convergencia y estabilidad.

DataSet Pre-training

Entrenamiento:

- Automotive.
- Cellphones and accessories.
- Clothing Shoes and Jewelry.
- Electronics
- Grocery and Gourmet Food
- Home and Kitchen
- Movies and TV

Validación:

- Cds and Vinyl

DataSet Finetuning

- Scientific.
- Instruments.
- Arts.
- Office
- Games
- Pet

Datasets	#Users	#Items	#Inters.	Avg. n	Density
Pre-training	3,613,906	1,022,274	33,588,165	9.29	9.1e-6
-Training	3,501,527	954,672	32,291,280	9.22	9.0e-6
-Validation	112,379	67,602	1,296,885	11.54	1.7e-4
Scientific	11,041	5,327	76,896	6.96	1.3e-3
Instruments	27,530	10,611	231,312	8.40	7.9e-4
Arts	56,210	22,855	492,492	8.76	3.8e-4
Office	101,501	27,932	798,914	7.87	2.8e-4
Games	11,036	15,402	100,255	9.08	5.9e-4
Pet	47,569	37,970	420,662	8.84	2.3e-4

Figura: Data set después del preprocesamiento

Comparación con baselines

Id-only methods:

- **GRU4Rec**: Mediante una RNN modela las secuencias de interacciones que corresponden a una sesión.
- **SASRec**: Usa un modelo self-attentive en una sola direccion, captura relaciones entre ítems dentro de la secuencia.
- **BERT4Rec**: Usa transformer bidireccional que tratar de predecir items ocultos en la secuencia.
- **RecGURU**: Pre entrena representaciones de secuencia usando autoencoders junto con aprendizaje advesarial.

Comparación con baselines

Id-text methods:

- **FDSA**: Modelo que usa auto-atención, capturando patrones de transición.
- **S3-Rec**: pre entrena el modelo secuencial, correlacionando atributos, items, subsecuencias, secuencias.

Text-only methods:

- **ZesRec**: Convierte el texto de los ítems en embeddings partir de un modelo de lenguaje pre entrenado.
- **UniSRec**: Representaciones textuales a partir de un modelo de lenguaje, usa adaptador MoE-enhance para ajustarse a nuevos dominios.

Rendimiento

Dataset	Metric	ID-Only Methods				ID-Text Methods		Text-Only Methods			Improv.
		GRU4Rec	SASRec	BERT4Rec	RecGURU	FDSA	S^3-Rec	ZESRec	UniSRec	RECFORMER	
Scientific	NDCG@10	0.0826	0.0797	0.0790	0.0575	0.0716	0.0451	0.0843	0.0862	**0.1027**	19.14%
	Recall@10	0.1055	0.1305	0.1061	0.0781	0.0967	0.0804	0.1260	0.1255	**0.1448**	10.96%
	MRR	0.0702	0.0696	0.0759	0.0566	0.0692	0.0392	0.0745	0.0786	**0.0951**	20.99%
Instruments	NDCG@10	0.0633	0.0634	0.0707	0.0468	0.0731	0.0797	0.0694	0.0785	**0.0830**	4.14%
	Recall@10	0.0969	0.0995	0.0972	0.0617	0.1006	0.1110	0.1078	**0.1119**	0.1052	-
	MRR	0.0707	0.0577	0.0677	0.0460	0.0748	0.0755	0.0633	0.0740	**0.0807**	6.89%
Arts	NDCG@10	0.1075	0.0848	0.0942	0.0525	0.0994	0.1026	0.0970	0.0894	**0.1252**	16.47%
	Recall@10	0.1317	0.1342	0.1236	0.0742	0.1209	0.1399	0.1349	0.1333	**0.1614**	15.37%
	MRR	0.1041	0.0742	0.0899	0.0488	0.0941	0.1057	0.0870	0.0798	**0.1189**	12.49%
Office	NDCG@10	0.0761	0.0832	0.0972	0.0500	0.0922	0.0911	0.0865	0.0919	**0.1141**	17.39%
	Recall@10	0.1053	0.1196	0.1205	0.0647	0.1285	0.1186	0.1199	0.1262	**0.1403**	9.18%
	MRR	0.0731	0.0751	0.0932	0.0483	0.0972	0.0957	0.0797	0.0848	**0.1089**	12.04%
Games	NDCG@10	0.0586	0.0547	0.0628	0.0386	0.0600	0.0532	0.0530	0.0580	**0.0684**	8.92%
	Recall@10	0.0988	0.0953	0.1029	0.0479	0.0931	0.0879	0.0844	0.0923	**0.1039**	0.97%
	MRR	0.0539	0.0505	0.0585	0.0396	0.0546	0.0500	0.0505	0.0552	**0.0650**	11.11%
Pet	NDCG@10	0.0648	0.0569	0.0602	0.0366	0.0673	0.0742	0.0754	0.0702	**0.0972**	28.91%
	Recall@10	0.0781	0.0881	0.0765	0.0415	0.0949	0.1039	0.1018	0.0933	**0.1162**	11.84%
	MRR	0.0632	0.0507	0.0585	0.0371	0.0650	0.0710	0.0706	0.0650	**0.0940**	32.39%

Figura: Performance de los distintos métodos

Resultados Clave

- Métodos ID-Text mejor desempeño que ID-Only Text-Only
- Recformer supera a todos los modelos comparativos(mejora 15.99 % en promedio), excepto en Recall en dataset de instrumentos.

Rendimiento en escenario Zero-shot

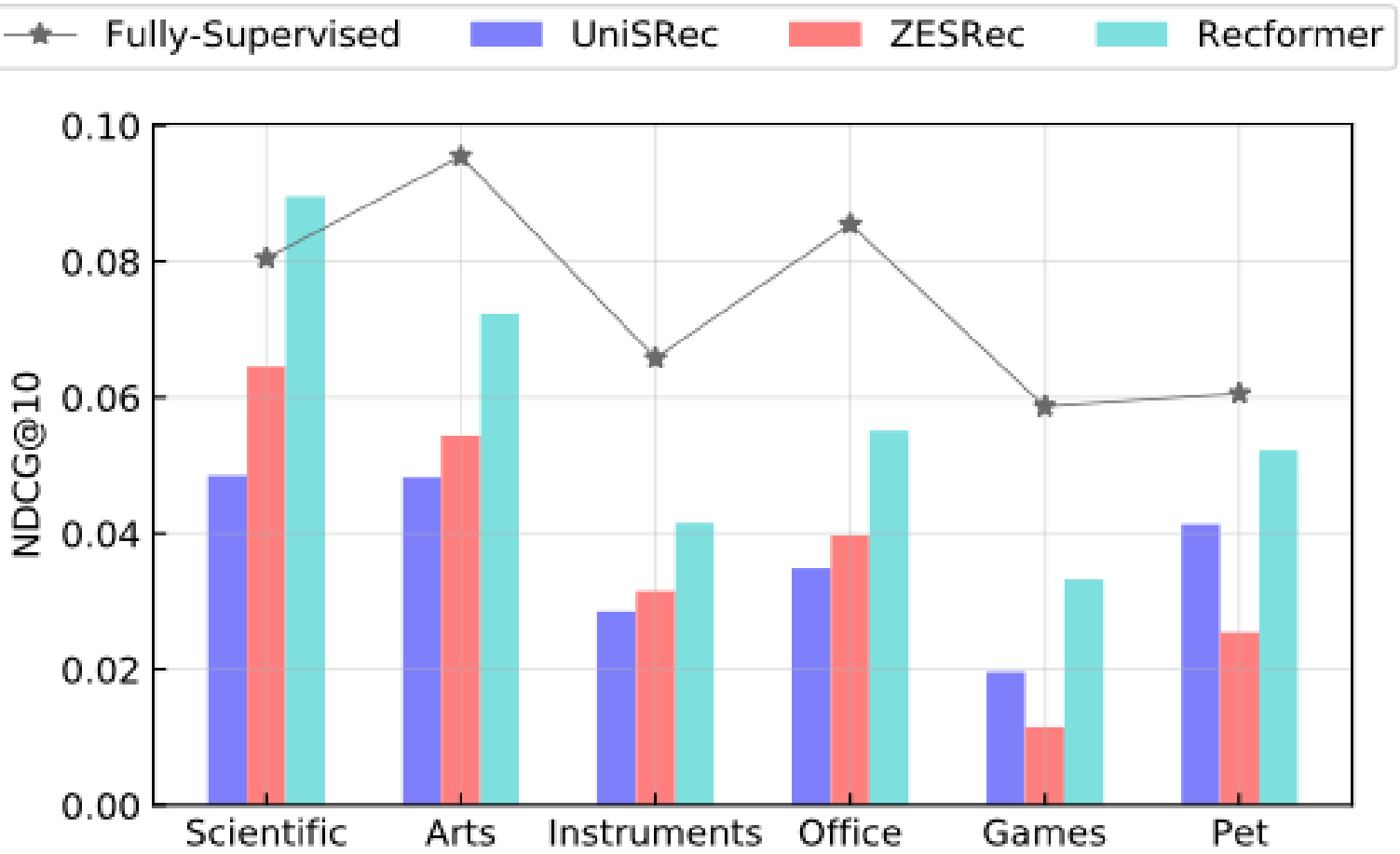

Figura: Rendimiento (NDCG@10) de tres métodos de solo texto en la configuración de disparo cero.

Rendimiento en escenario Zero-shot

- Evalúa modelos Text-Only (Recformer, UniSRec, ZESRec) sin fine-tuning.
- Recformer supera incluso a métodos ID-Only totalmente entrenados en algunos casos.
- Demuestra la capacidad de transferencia sin entrenamiento adicional.

Rendimiento en escenario con pocos datos

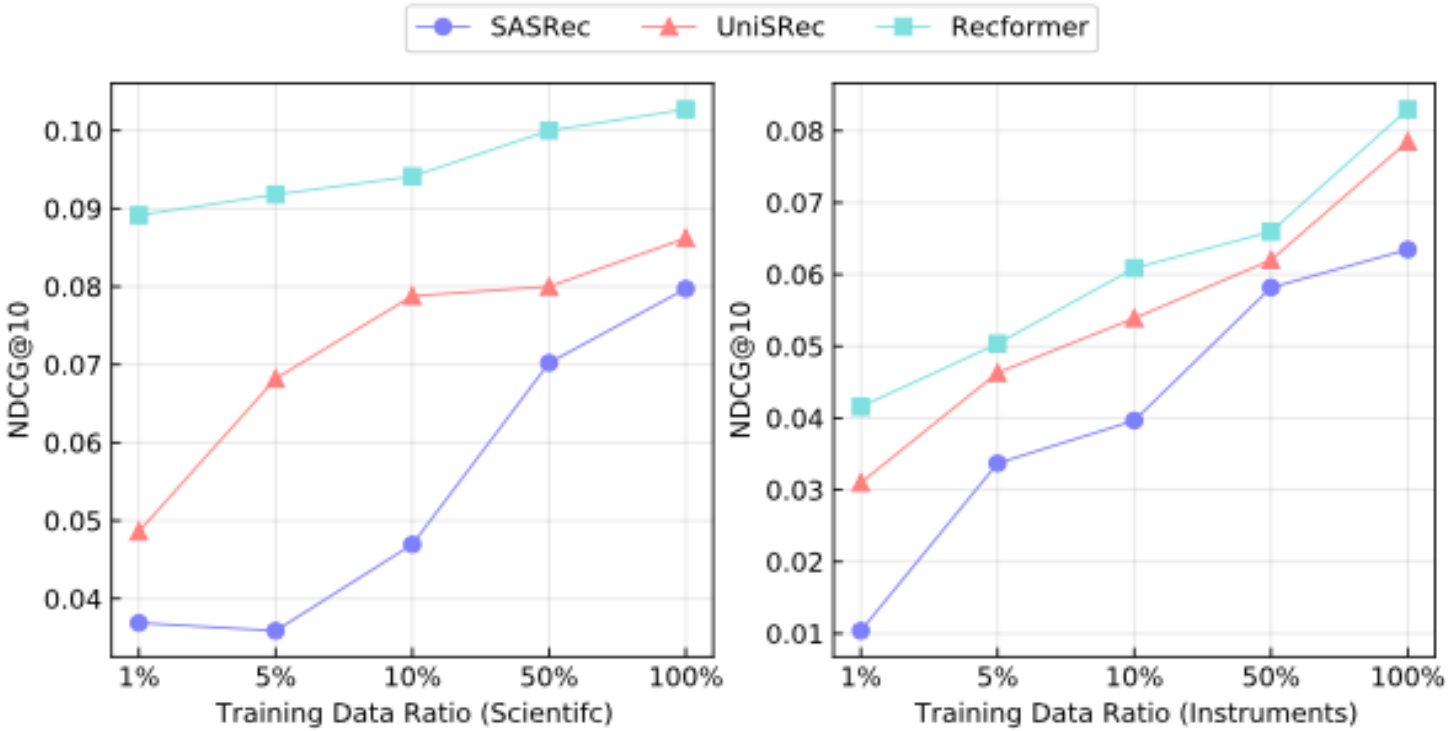

Figura: Rendimiento (NDCG@10) de SASRec, UniSRec y Recformer en diferentes tamaños (es decir, 1 %, 5 %, 10 %, 50 %, 100 %) de datos de entrenamiento.

Rendimiento en escenario con pocos datos

- Compara el rendimiento con diferentes proporciones de datos (1 %, 5 %, 10 %, etc.).
- Recformer y UniSRec superan a SASRec especialmente cuando hay poca data.
- Recformer es más robusto en escenarios de escasez de datos.

Ítems Cold-Start

		SASRec		UniSRec		RECFORMER	
Dataset	**Metric**	**In-Set**	**Cold**	**In-Set**	**Cold**	**In-Set**	**Cold**
Scientific	N@10	0.0775	0.0213	0.0864	0.0441	0.1042	0.0520
	R@10	0.1206	0.0384	0.1245	0.0721	0.1417	0.0897
Instruments	N@10	0.0669	0.0142	0.0715	0.0208	0.0916	0.0315
	R@10	0.1063	0.0309	0.1094	0.0319	0.1130	0.0468
Arts	N@10	0.1039	0.0071	0.1174	0.0395	0.1568	0.0406
	R@10	0.1645	0.0129	0.1736	0.0666	0.1866	0.0689
Pet	N@10	0.0597	0.0013	0.0771	0.0101	0.0994	0.0225
	R@10	0.0934	0.0019	0.1115	0.0175	0.1192	0.0400

Figura: Rendimiento de modelos comparado entre elementos dentro del conjunto y elementos de inicio en frío en cuatro conjuntos de datos

Ítems Cold-Start

- Recformer supera ampliamente a métodos basados en IDs (como SASRec) en ítems nunca vistos.
- Ventaja clara de usar textos frente a embeddings de ítems preentrenados.

Estudio de Ablación

Variants	Scientific			Instruments		
	NDCG@10	Recall@10	MRR	NDCG@10	Recall@10	MRR
(0) RECFORMER	**0.1027**	**0.1448**	**0.0951**	**0.0830**	**0.1052**	**0.0807**
(1) w/o two-stage finetuning	0.1023	0.1442	0.0948	0.0728	0.1005	0.0685
(1) + (2) freezing word emb. & item emb.	0.1026	0.1399	0.0942	0.0728	0.1015	0.0682
(1) + (3) trainable word emb. & item emb.	0.0970	0.1367	0.0873	0.0802	0.1015	0.0759
(1) + (4) trainable item emb. & freezing word emb.	0.0965	0.1383	0.0856	0.0801	0.1014	0.0760
(5) w/o pre-training	0.0722	0.1114	0.0650	0.0598	0.0732	0.0584
(6) w/o item position emb. & token type emb.	0.1018	0.1427	0.0945	0.0518	0.0670	0.0501

Figura: Estudio de ablación en dos datasets

Estudio de Ablación

- Se estudian variantes del modelo para ver el impacto de cada componente:
  - Fine-tuning en dos etapas mejora el rendimiento.
  - Pre-entrenamiento es crucial.
  - Embeddings de posición y tipo de token también son importantes.

Pasos de pre-entrenamiento

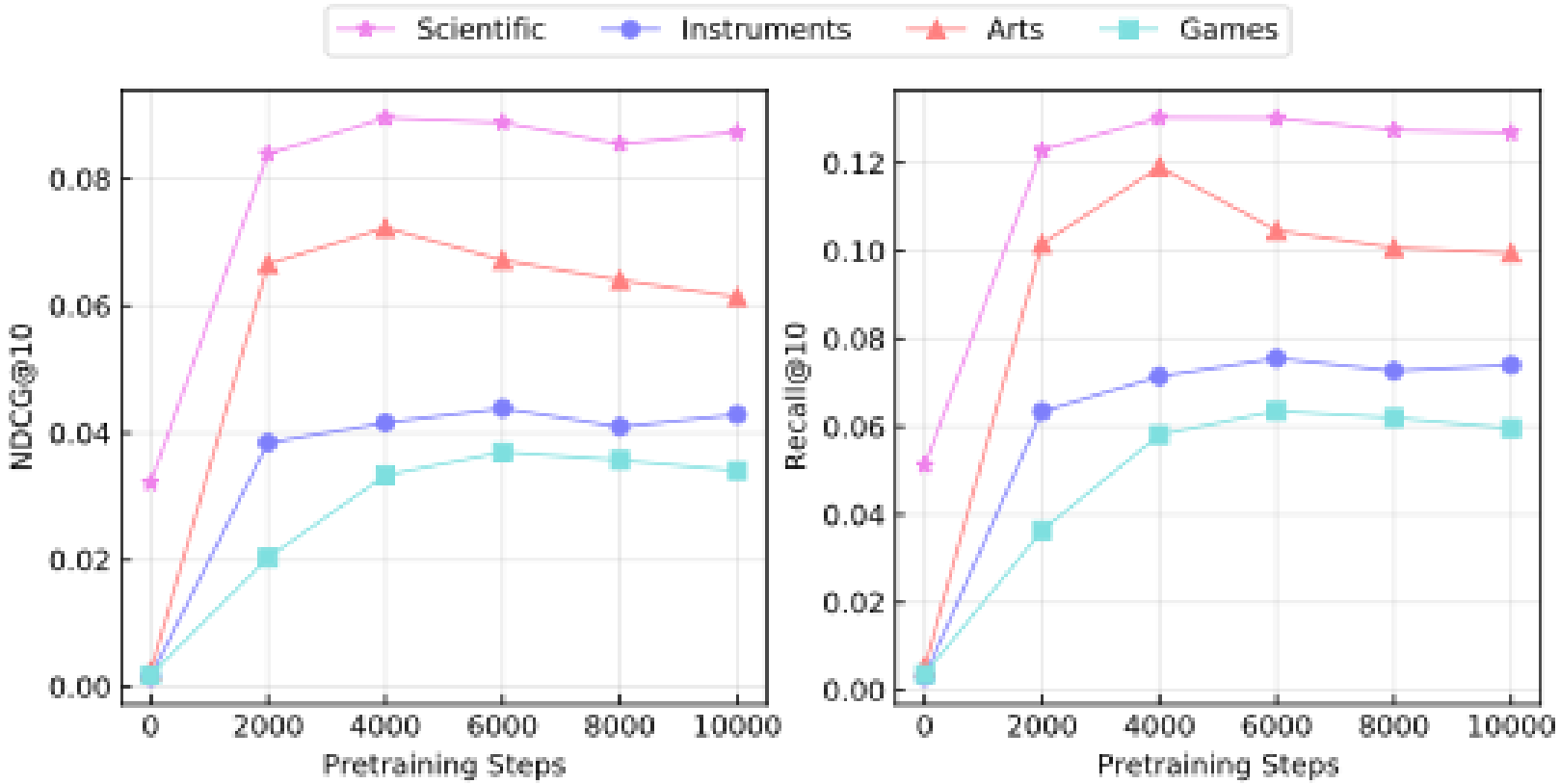

Figura: Rendimiento de la recomendación de disparo cero del Recformador (NDCG@10 y Recall@10) en diferentes pasos previos al entrenamiento.

Fortalezas de Recformer

- Generalización sin necesidad de IDs.
- Representaciones transferibles entre dominios.
- Escalabilidad sin tablas de embeddings.

Debilidades y Desafíos

- Alto costo computacional (modelo de lenguaje).
- Requiere metadatos textuales ricos.
- Menor aplicabilidad en dominios sin descripciones.

Conclusiones

- Recformer unifica lenguaje natural y recomendación secuencial.
- Alta capacidad de transferencia.
- Rompe con la dependencia de IDs y tablas estáticas.

Referencias

- Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. ArXiv abs/1607.06450 (2016).
- Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. ArXiv abs/2004.05150 (2020).
- Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On Sampling Strategies for Neural Network-based Collaborative Filtering. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2017).

Referencias (1/4)

1. Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. ArXiv abs/1607.06450 (2016).
2. Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. ArXiv abs/2004.05150 (2020).
3. Ting Chen, Yizhou Sun, Yue Shi, and Liangjie Hong. 2017. On Sampling Strategies for Neural Network-based Collaborative Filtering. KDD (2017).
4. Yong-Guang Chen et al. 2022. Intent Contrastive Learning for Sequential Recommendation. WWW (2022).
5. Junyoung Chung et al. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. ArXiv abs/1412.3555 (2014).
6. Jacob Devlin et al. 2019. BERT: Pre-training of Deep Bidirectional Transformers. ArXiv abs/1810.04805 (2019).
7. Hao Ding et al. 2021. Zero-Shot Recommender Systems. ArXiv abs/2105.08318 (2021).
8. Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE. EMNLP.
9. Ruining He and Julian McAuley. 2016. Fusing Similarity Models with Markov Chains. ICDM (2016).
10. Dan Hendrycks and Kevin Gimpel. 2016. Gaussian Error Linear Units (GELUs). ArXiv.

Referencias (2/4)

11. Balázs Hidasi et al. 2015. Session-based Recommendations with RNNs. CoRR abs/1511.06939.
12. Yupeng Hou et al. 2022. Towards Universal Sequence Representation. KDD (2022).
13. Guangneng Hu et al. 2018. CoNet. CIKM (2018).
14. Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. ICDM (2018).
15. Mike Lewis et al. 2019. BART. ACL.
16. Chenglin Li et al. 2021. RecGURU. WSDM (2021).
17. Jing Li et al. 2017. Neural Attentive Session-based Recommendation. CIKM (2017).
18. Jiacheng Li et al. 2022. UCTopic. ACL 2022.
19. Jiacheng Li et al. 2020. Time Interval Aware Self-Attention. WSDM (2020).
20. Jiacheng Li et al. 2022. Coarse-to-Fine Sparse Sequential Recommendation. SIGIR (2022).

Referencias (3/4)

21. Yinhan Liu et al. 2019. RoBERTa. ArXiv abs/1907.11692.
22. Jianmo Ni et al. 2019. Justifying Recommendations. EMNLP.
23. Alec Radford and Karthik Narasimhan. 2018. GPT Pre-Training.
24. Colin Raffel et al. 2019. Exploring the Limits of Transfer Learning. ArXiv abs/1910.10683.
25. Steffen Rendle et al. 2010. Factorizing Markov chains. WWW.
26. Ajit Paul Singh and Geoffrey J. Gordon. 2008. Collective Matrix Factorization. KDD.
27. Fei Sun et al. 2019. BERT4Rec. CIKM.
28. Jiaxi Tang and Ke Wang. 2018. Convolutional Sequence Embedding. WSDM.
29. Jie Tang et al. 2012. Cross-domain collaboration recommendation. KDD.
30. Ashish Vaswani et al. 2017. Attention is All You Need. ArXiv abs/1706.03762.

Referencias (4/4)

31. Chuhan Wu et al. 2020. PTUM. ArXiv abs/2010.01494.
32. Chaojun Xiao et al. 2021. UPRec. ArXiv abs/2102.10989.
33. Ruobing Xie et al. 2021. Contrastive Cross-domain Recommendation. KDD.
34. Fajie Yuan et al. 2018. Convolutional Generative Network. WSDM.
35. Fajie Yuan et al. 2020. Continual User Representation. SIGIR.
36. Manzil Zaheer et al. 2020. Big Bird. ArXiv abs/2007.14062.
37. Tingting Zhang et al. 2019. Feature-level Self-Attention. IJCAI.
38. Kun Zhou et al. 2020. S3-Rec. CIKM.
39. Feng Zhu et al. 2019. DTCDR. CIKM.
40. Feng Zhu et al. 2021. Cross-Domain Recommendation Survey. ArXiv abs/2103.01696.