



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DE CHILE

Departamento de Ciencia de la Computación  
Escuela de Ingeniería

# Effective and Efficient Training for Sequential Recommendation using Recency Sampling

**Autores:** Aleksandr Petrov & Craig Macdonald

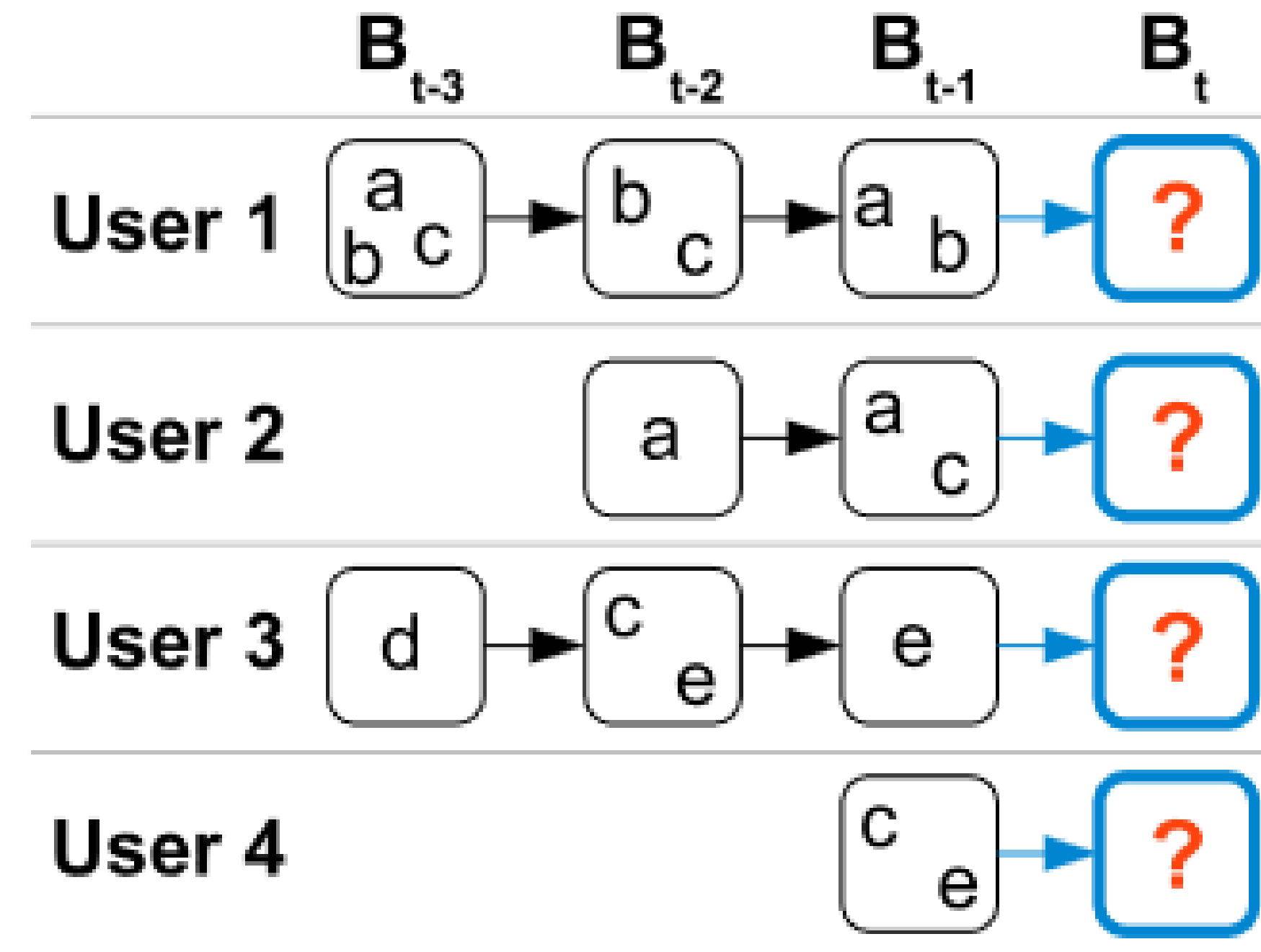
**Estudiantes:** Gabriel Astudillo Laroze  
Miguel Fernández Pizarro  
Javier Ramos Di Consoli  
Ariel Reyes Pardo

02 de noviembre de 2023

# 1. Introducción

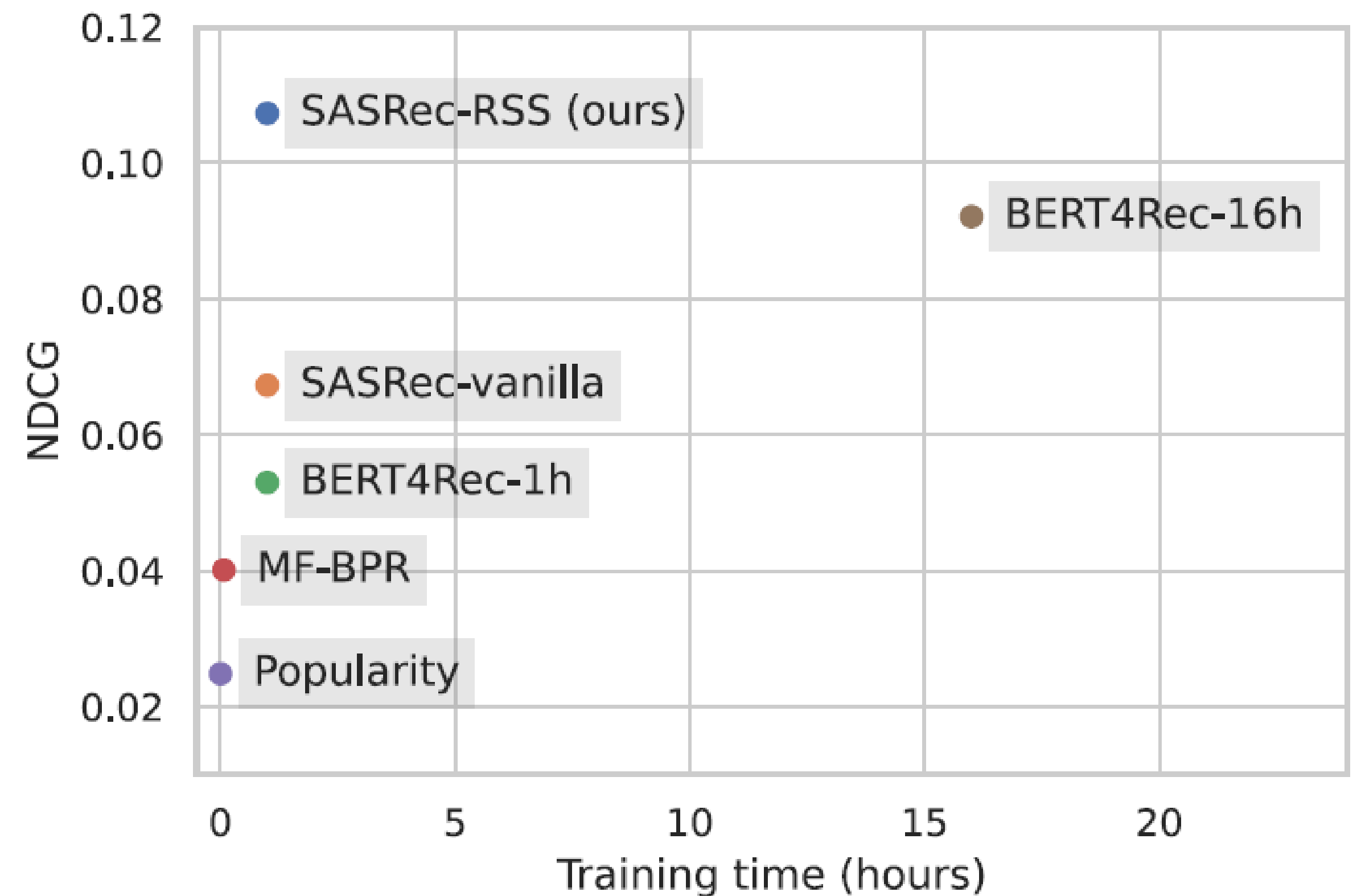
El enfoque secuencial en sistemas recomendadores toma en consideración el orden en las interacciones entre usuarios e ítems. Uno de los primeros esfuerzos consideraba cadenas de Markov.

Modelos actuales se basan principalmente en arquitecturas de aprendizaje profundo como redes recurrentes y modelos transformers.



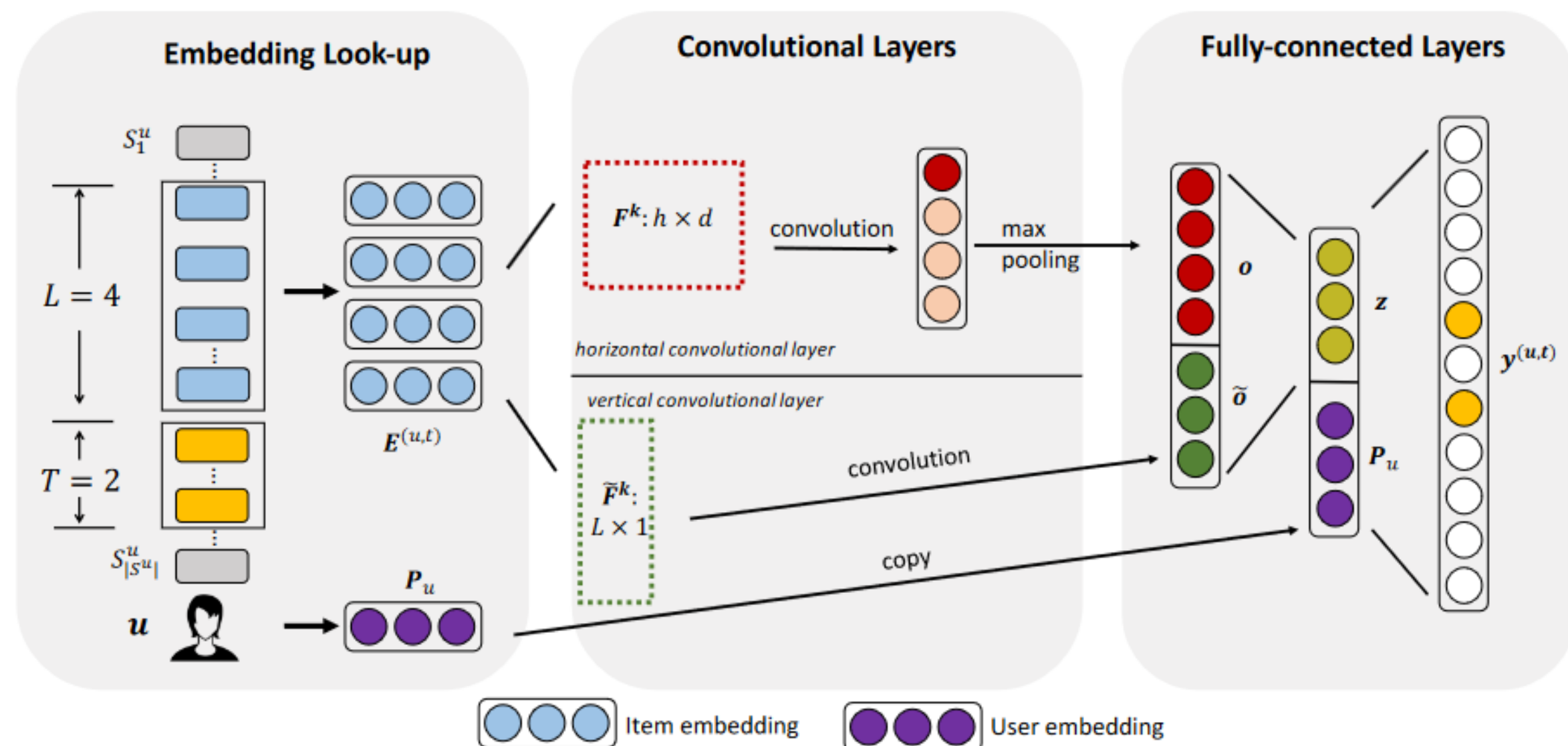
# 1. Introducción

Las arquitecturas de Deep learning, utilizadas en el contexto de sistemas recomendadores, pueden obtener buenos resultados al estimar relevancia, pero el tiempo de entrenamiento es alto. Este hecho limita la posibilidad actualizar el modelo según las nuevas preferencias de los usuarios.

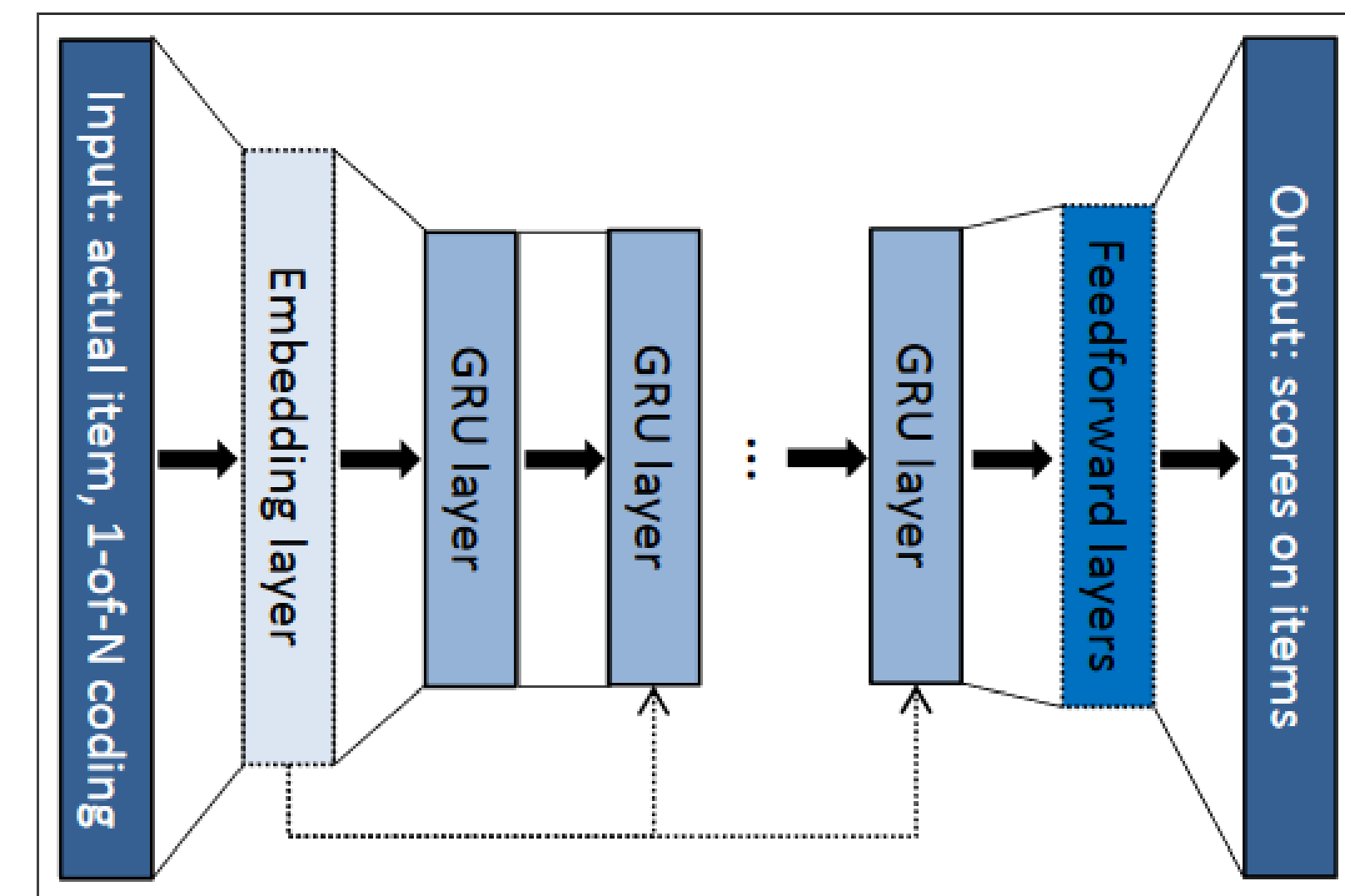


## 2. Trabajos relacionados

Caser (1)



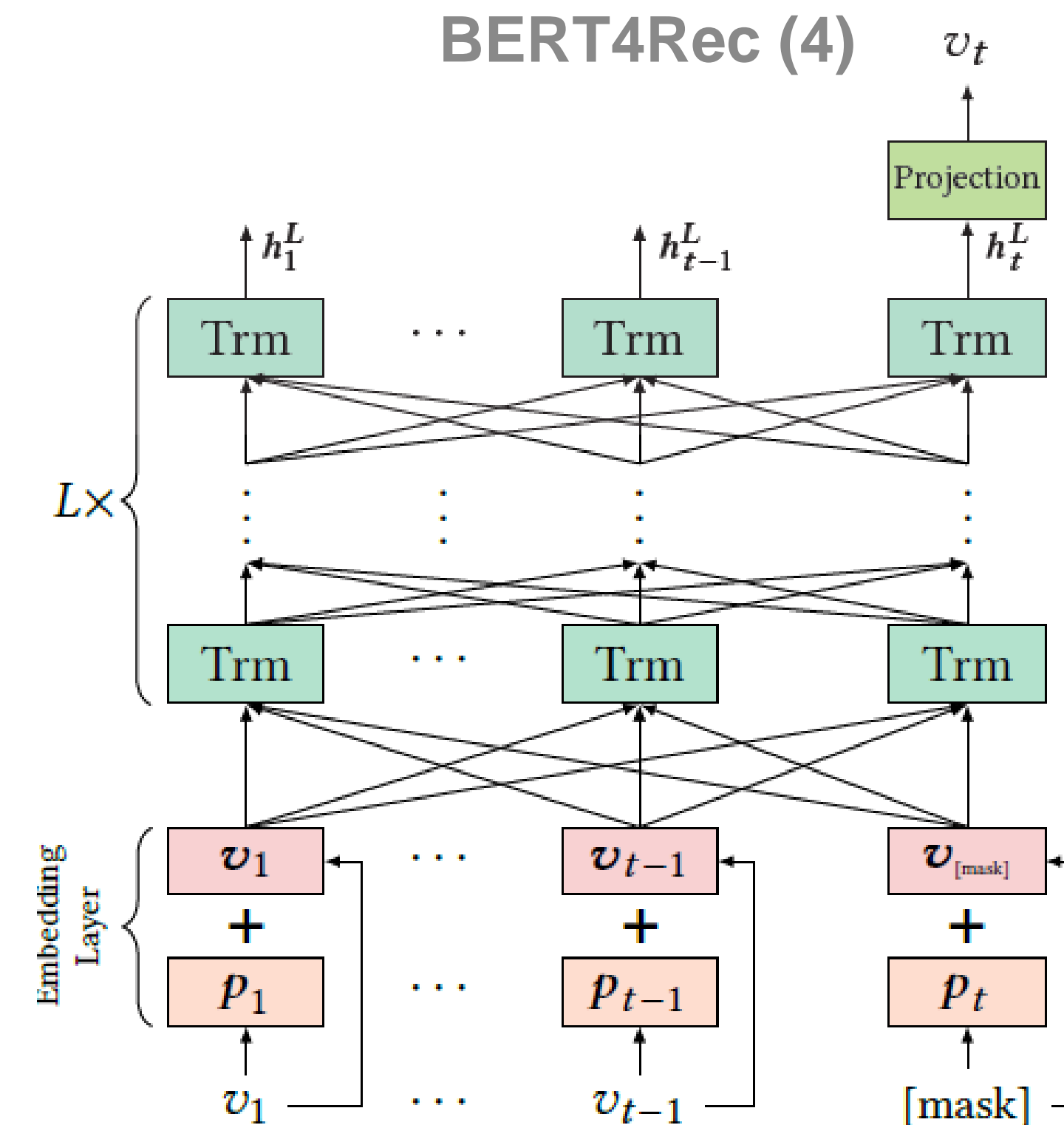
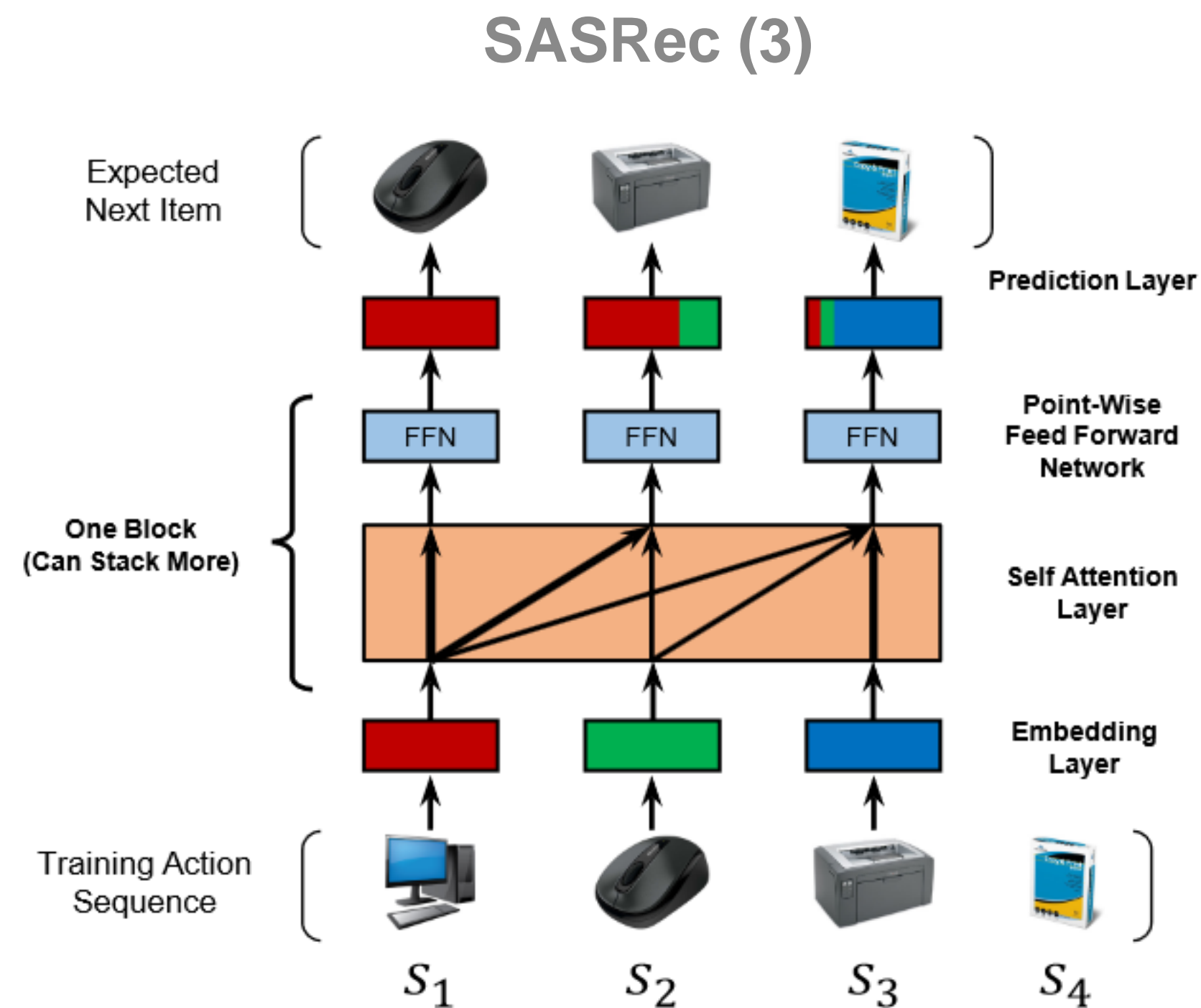
GRU4Rec (2)



(1) Tang, J., & Wang, K. (2018). **Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding**. Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining.

(2) Hidasi, B., Karatzoglou, A., Baltrunas, L., & Tikk, D. (2015). **Session-based Recommendations with Recurrent Neural Networks**. CoRR, abs/1511.06939.

## 2. Trabajos relacionados



Kang, W., & McAuley, J. (2018). **Self-Attentive Sequential Recommendation**. 2018 IEEE International Conference on Data Mining (ICDM), 197-206.

(4) Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, P. (2019). **BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer**. Proceedings of the 28th ACM International Conference on Information and Knowledge Management..

## 2. Trabajos relacionados

De los modelos presentados, se reconocen 2 limitaciones:

### Limitación 1

La estrategia “Sequence continuation” solo puede generar un pequeño número de muestras de entrenamiento a partir de una secuencia. Esto permite que el entrenamiento se realice relativamente rápido, pero el rendimiento de estos modelos es inferior en comparación con BERT4Rec.

### Limitación 2

La reconstrucción de elementos “enmascarados” es una tarea muy general y está débilmente relacionada con la tarea de recomendación secuencial. Bajo este enfoque, los modelos pueden alcanzar un buen rendimiento, pero el tiempo de entrenamiento es mayor que otros objetivos de entrenamiento.



### 3. Propuesta de solución: Recency-based Sampling of Sequences (RSS)

- Cada element de la secuencia puede ser seleccionado. Usando este principio, se logra generar hasta  $2^n$  muestras de entrenamiento a partir de cada secuencia de usuario.

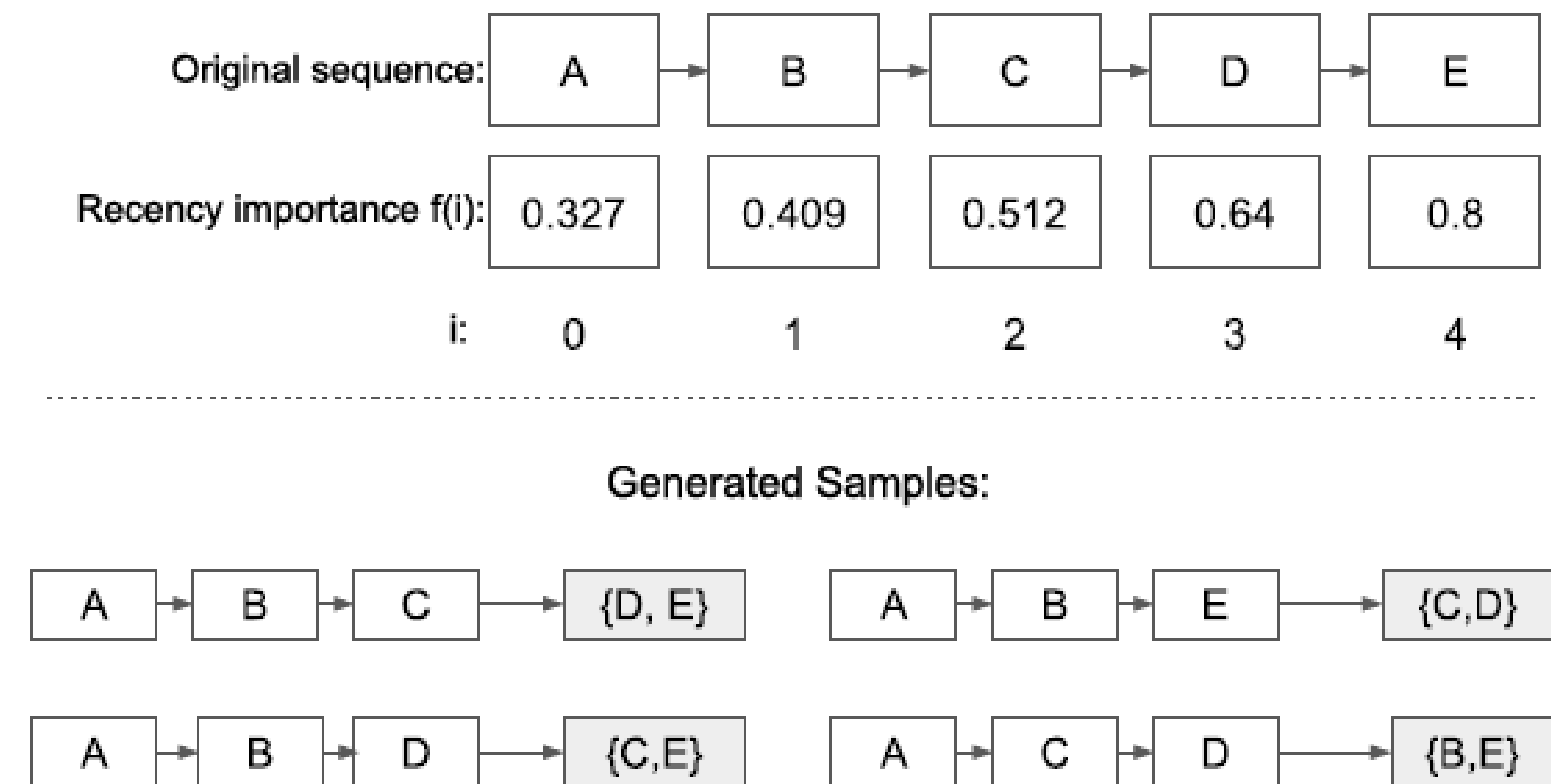
**Este principio aborda la Limitación 1.**

- Las interacciones más recientes en una secuencia indican mejor los intereses del usuario, y por lo tanto, son objetivos más realistas. Por lo tanto, los métodos que se basan en este principio mantendrán una estrecha relación con las recomendaciones secuenciales.

**Este principio aborda la Limitación L2.**

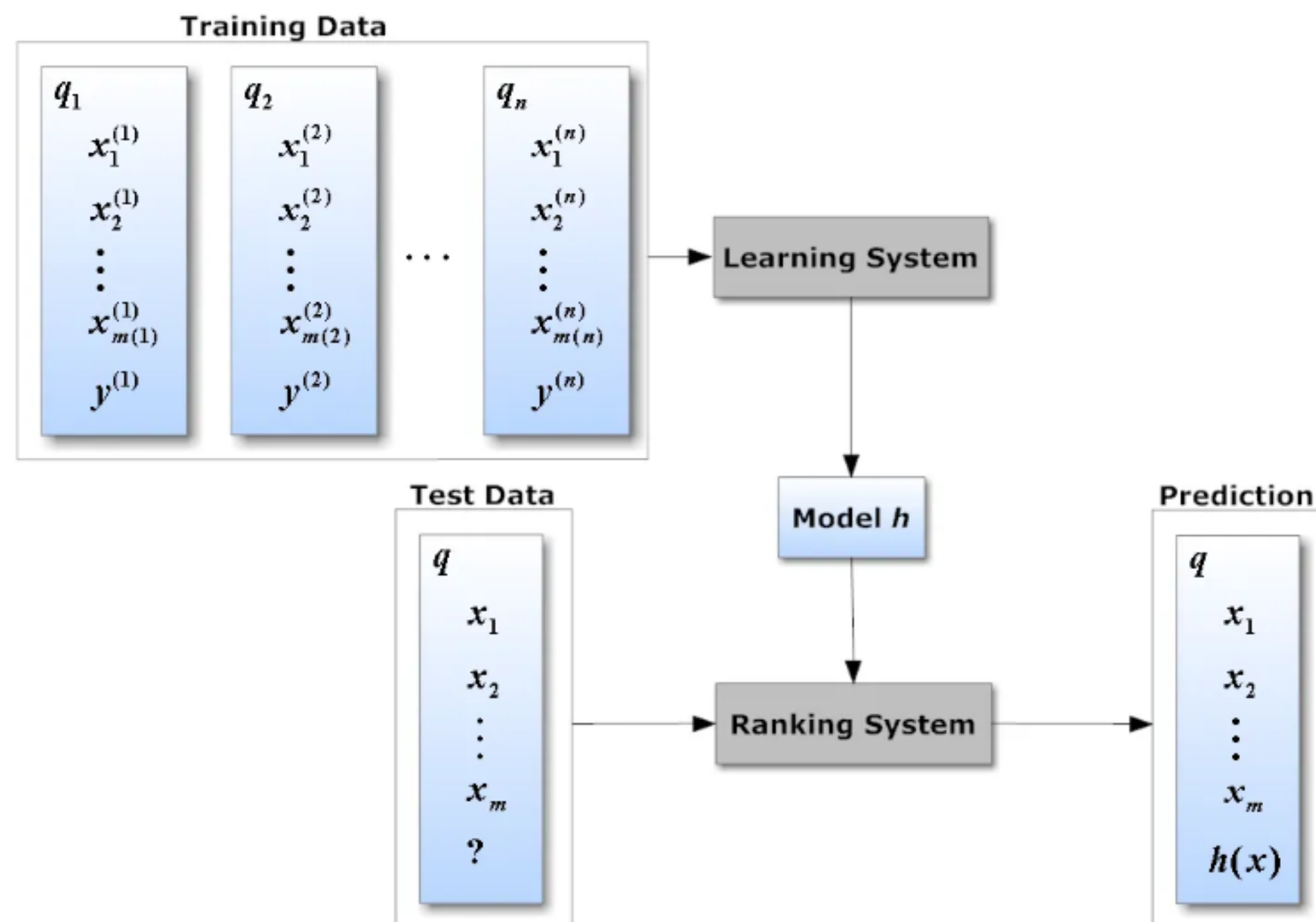
$$p(i) = \frac{f(i)}{\sum_{j=0}^{n-1} f(j)} \quad (1)$$

$$f(k) = \alpha^{n-k} \quad (2)$$



### 3. Propuesta de solución: Función de pérdida $\lambda$ Rank

RSS usa LambdaRank ( $\lambda$ Rank) (5)



$$\lambda_i = \sum_{j \in I} |\Delta NDCG_{ij}| \frac{-\sigma}{1 + e^{\sigma(s_i - s_j)}} \quad (3)$$



En este contexto, el gradiente se puede interpretar como flechas que indican hacia donde se deben mover los ítems para alcanzar la posición correcta.

(5) Burges, C. J. C. (2010). From RankNet to LambdaRank to LambdaMART: An Overview. Microsoft Research.



## 4. Experimentación y resultados

Las preguntas de investigación son 4:

**RQ1:** ¿Recency-based Sampling of Sequences ayuda en el entrenamiento de modelos secuenciales?

**RQ2:** ¿La función de pérdida  $\lambda$ Rank beneficia el proceso de entrenamiento de RSS?

**RQ3:** ¿Cuál es el impacto del parámetro  $\alpha$  (recency importance) en la función exponencial de recency?

**RQ4:** Comparado con los baselines que son el estado del arte, ¿RSS arroja buenos resultados?

## 4. Experimentación y resultados

Se consideran 4 datasets:

Name	Users	Items	Interactions	Average length	Median length	sparsity
Booking.com	140746	34742	917729	6.52	6	0.999812
Gowalla	86168	1271638	6397903	74.24	28	0.999942
Yelp	287116	148523	4392169	15.29	8	0.999897
MovieLens-20M	138493	26744	20000263	144.413530	68	0.994600

Booking.com → Destinos de viajes

Gowalla → Ubicaciones

Yelp → Negocios

MovieLens-20M → Películas

## 4. Experimentación y resultados

Evaluación de métricas Recall@10 y NDCG@10:

(a) Recall@10

		MovieLens-20M		Yelp		Gowalla		Booking.com	
Architecture	Loss	Cont	RSS	Cont	RSS	Cont	RSS	Cont	RSS
GRU4Rec	BCE	0.0221†	<b>0.0354*</b>	0.0075†	<b>0.0100*†</b>	<b>0.0026*</b>	0.0005	0.4621	<b>0.4962*</b>
	$\lambda$ Rank	0.0082	<b>0.1544*†</b>	0.0009	<b>0.0045*</b>	0.0068†	<b>0.0119*†</b>	<b>0.4780†</b>	<b>0.5084*†</b>
Caser	BCE	0.1424†	<b>0.1866*</b>	0.0046†	<b>0.0099*†</b>	0.0076	<b>0.0081</b>	<b>0.5600*†</b>	0.5454†
	$\lambda$ Rank	0.0330	<b>0.1496*†</b>	0.0009	<b>0.0017*</b>	0.0087†	<b>0.0157*†</b>	0.4968	<b>0.5273*</b>
SASRec	BCE	0.1537†	<b>0.1888*</b>	0.0146†	<b>0.0269*†</b>	0.0089	0.0089	<b>0.5845*†</b>	0.5178
	$\lambda$ Rank	0.1050	<b>0.1968*†</b>	0.0045	<b>0.0052*</b>	0.0715	<b>0.1020*†</b>	<b>0.5662*</b>	0.52464†

(b) NDCG@10

		MovieLens-20M		Yelp		Gowalla		Booking.com	
Architecture	Loss	Cont	RSS	Cont	RSS	Cont	RSS	Cont	RSS
GRU4Rec	BCE	0.0115†	<b>0.0183*</b>	0.0035†	<b>0.0049*†</b>	<b>0.0017*</b>	0.0002	0.2829	<b>0.2899*</b>
	$\lambda$ Rank	0.0040	<b>0.0839*†</b>	0.0004	<b>0.0014*</b>	0.0033†	<b>0.0067*†</b>	<b>0.3132*†</b>	<b>0.3093†</b>
Caser	BCE	0.0784†	<b>0.0995*</b>	0.0021†	<b>0.0049*†</b>	0.0039	<b>0.0040</b>	<b>0.3665*†</b>	0.3311†
	$\lambda$ Rank	0.0177	<b>0.0814*†</b>	0.0003	<b>0.0007*</b>	0.0055†	<b>0.0100*†</b>	0.3181	<b>0.3226*</b>
SASRec	BCE	0.0850†	<b>0.1002*</b>	0.0076†	<b>0.0136*†</b>	0.0044	<b>0.0044</b>	<b>0.3633*†</b>	0.2966
	$\lambda$ Rank	0.0579	<b>0.1073*†</b>	0.0021	<b>0.0025*</b>	0.0478†	<b>0.0749*†</b>	<b>0.3623*</b>	0.3122†

## 4. Experimentación y resultados

Controlando por tiempo de entrenamiento:

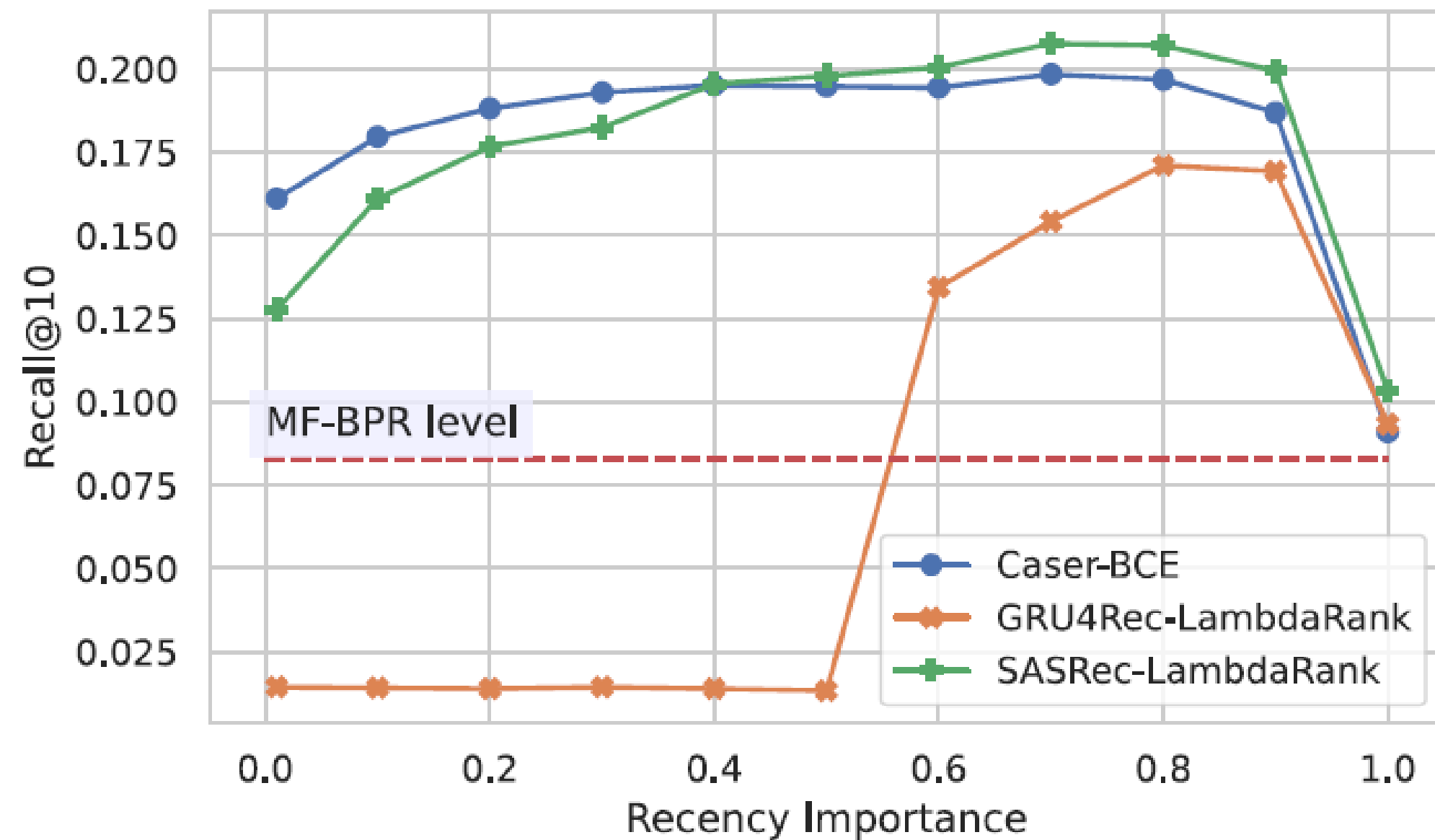
Model	Train time	MovieLens-20M		Yelp		Gowalla		Booking.com	
		Recall @10	NDCG @10	Recall @10	NDCG @10	Recall @10	NDCG @10	Recall @10	NDCG @10
Popularity	1h	0.049 <sup>†*</sup>	0.025 <sup>†*</sup>	0.006 <sup>†</sup>	0.003 <sup>†*</sup>	0.008 <sup>*</sup>	0.004 <sup>*</sup>	0.097 <sup>†*</sup>	0.043 <sup>†*</sup>
MF-BPR	1h	0.079 <sup>†*</sup>	0.040 <sup>†*</sup>	0.019 <sup>†*</sup>	0.009 <sup>†*</sup>	<u>0.029<sup>†*</sup></u>	<u>0.018<sup>†*</sup></u>	0.449 <sup>†*</sup>	0.279 <sup>†*</sup>
SASRec-vanilla	1h	0.136 <sup>†*</sup>	0.067 <sup>†*</sup>	<u>0.022<sup>†*</sup></u>	<u>0.011<sup>†*</sup></u>	0.010 <sup>*</sup>	0.005 <sup>†*</sup>	0.463 <sup>†*</sup>	0.270 <sup>†*</sup>
BERT4rec-1h	1h	0.107 <sup>†*</sup>	0.053 <sup>†*</sup>	0.014 <sup>†*</sup>	0.007 <sup>†*</sup>	N/A <sup>1</sup>	N/A <sup>1</sup>	0.479 <sup>†*</sup>	0.288 <sup>†*</sup>
SASRec-RSS-BCE	1h	<u>0.189<sup>*</sup></u>	<u>0.100<sup>*</sup></u>	<b>0.027<sup>*</sup></b>	<b>0.014<sup>*</sup></b>	0.009 <sup>*</sup>	0.004 <sup>*</sup>	<u>0.518<sup>*</sup></u>	<u>0.297<sup>*</sup></u>
SASRec-RSS- $\lambda$ Rank	1h	<b>0.197<sup>†</sup></b>	<b>0.107<sup>†</sup></b>	0.005 <sup>†</sup>	0.003 <sup>†</sup>	<b>0.102<sup>†</sup></b>	<b>0.075<sup>†</sup></b>	<b>0.525<sup>†</sup></b>	<b>0.312<sup>†</sup></b>
BERT4Rec-16h <sup>2</sup>	16h	0.173 <sup>†*</sup>	0.092 <sup>†*</sup>	0.028 <sup>*</sup>	0.014 <sup>*</sup>	N/A <sup>1</sup>	N/A <sup>1</sup>	0.565 <sup>†*</sup>	0.354 <sup>†*</sup>

**Bold** denotes the best model for a dataset by the metric in the main group, underlined the second best. Symbols \* and † denote statistically significant difference compared with SASRec-RSS-BCE and SASRec-RSS- $\lambda$ Rank respectively, according to a paired t-test with Bonferroni multiple testing correction ( $pvalue < 0.05$ ).

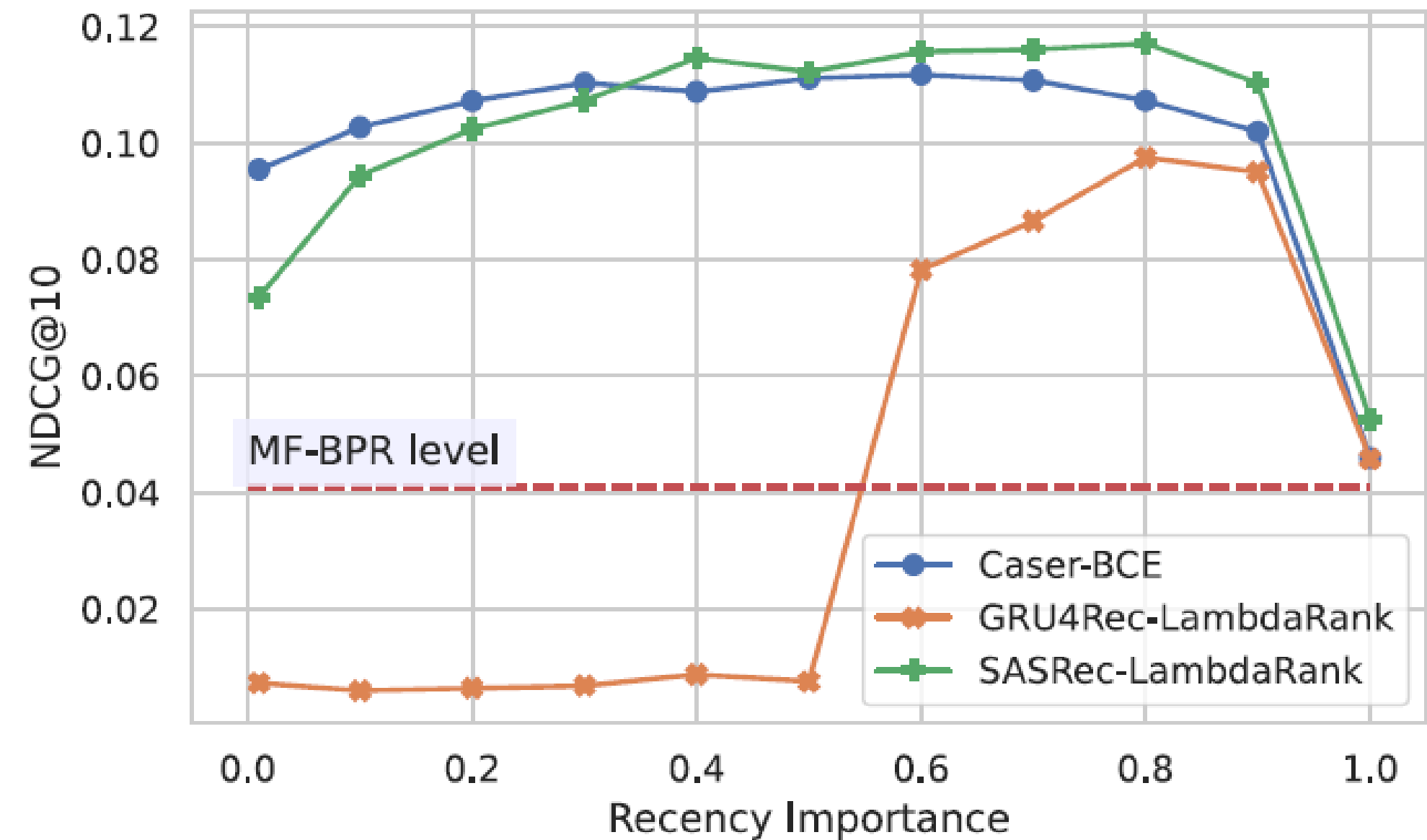
<sup>1</sup> We do not report results for BERT4Rec models for the Gowalla dataset because due to large number of items in this dataset, we were not able to train the model. <sup>2</sup> We report results for BERT4rec-16h separately due to its larger training time.

## 4. Experimentación y resultados

En los siguientes gráficos se ve el efecto al variar  $\alpha$  (recency importance):



(a) Effect of position recency importance on Recall



(b) Effect of position recency importance on NDCG

## 5. Conclusiones

**RQ1:** Los autores concluyen que RSS mejora el entrenamiento de modelos si los ítems en la secuencia del usuario pueden tratarse como target.

**RQ2:** En general,  $\lambda$ Rank mejora el rendimiento de los modelos.

**RQ3:** La variación del parámetro de  $\alpha$  (recency importance) tiene un impacto significativo en el rendimiento de los modelos.

**RQ4:** El método RSS, específicamente a través de SASRec-RSS, supera significativamente a los modelos de referencia en todos los conjuntos de datos evaluados dentro del límite de tiempo de entrenamiento.



## 6. Crítica

- Los autores argumentan que la función de pérdida lambda rank produce una mejora en los resultados, lo que no parece tan evidente. Esta situación varía a través de las distintas arquitecturas y dataset.
- La publicación pone como foco central la relación entre eficiencia y eficacia. Este hecho nos parece destacable.
- Sería positivo desarrollar una evaluación sobre el rendimiento y los requisitos computacionales de RSS sobre datasets de mayor volumen.



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DE CHILE

Departamento de Ciencia de la Computación  
Escuela de Ingeniería

# Effective and Efficient Training for Sequential Recommendation using Recency Sampling

**Autores:** Aleksandr Petrov & Craig Macdonald

**Estudiantes:** Gabriel Astudillo Laroze  
Miguel Fernández Pizarro  
Javier Ramos Di Consoli  
Ariel Reyes Pardo

02 de noviembre de 2023

# ANEXOS 2: Pseudocódigo de la propuesta

---

**Algorithm 1** Recency-based Sampling of Sequences

---

**Input:** *sequence* - a sequence of interactions;  $\tau$  - maximum percent of target items;  $f$  - recency importance function

**Output:** *input* is a generated input sequence for the model; *target* is a set of sampled positive items

**function** RECENCYSEQUENCESAMPLING(*sequence*,  $\tau$ ,  $f$ )

*sampledIdx*  $\leftarrow$  set()

$n \leftarrow \text{length}(\text{sequence})$ ;  $c \leftarrow \max(1, \text{int}(n * \tau))$

*prob*  $\leftarrow$  Array[ $n$ ]

*prob*[ $i$ ]  $\leftarrow \frac{f(i)}{\sum_{j=0}^{n-1} f(j)}$  **for**  $i$  **in**  $[0, n - 1]$

*sampledIdx*  $\leftarrow$  random.choice(range( $0..n - 1$ ),  $c$ , *prob*)

*input*  $\leftarrow$  list(); *target*  $\leftarrow$  set()

**for**  $i \leftarrow 0, n - 1$  **do**

**if**  $i \in \text{sampledIdx}$  **then** *target.add(sequence[i])* **else**  
        *input.append(sequence[i])*

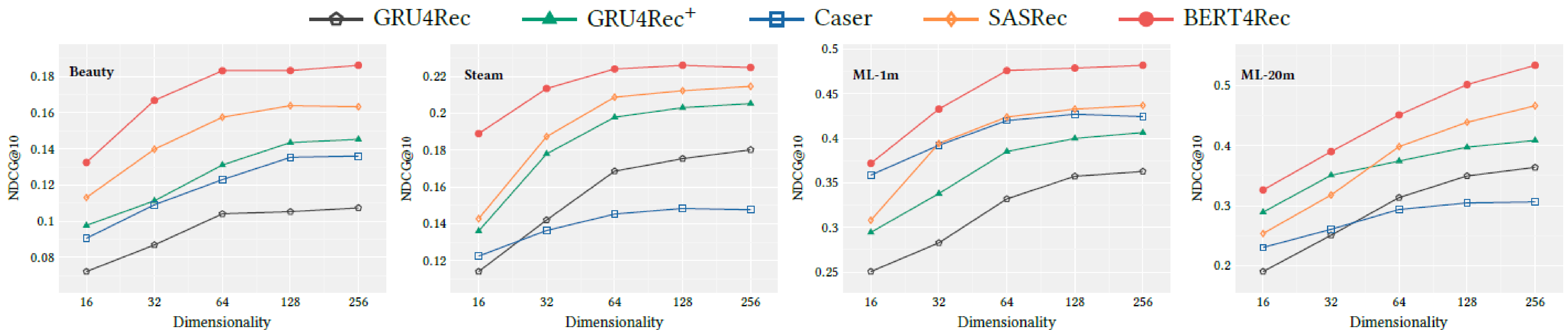
**end for**

**return** *input*, *target*

**end function**

# ANEXO 1: Comparación modelos en BERT4Rec

- Comparación de resultados (5)



(5) Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., & Jiang, P. (2019). **BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer**. Proceedings of the 28th ACM International Conference on Information and Knowledge Management.