Detección Estadística del Sesgo Posicional en Sistemas de Recomendación basados en Grandes Modelos de Lenguaje

Carlos Peláez-González[1 [0000−0001−7976−3139]],
Andrés Herrera-Poyatos[2 [0000−0001−9183−5262]], y
Francisco Herrera[1 [0000−0002−7283−312X]]

[1] Departamento de Ciencias de la Computación e Inteligencia Artificial, Universidad de Granada, Granada (18010), España
carlosprog@ugr.es
herrera@decsai.ugr.es
[2] Departamento de Álgebra, Universidad de Granada, Granada (18010), España
andreshp@ugr.es

Resumen Los sistemas de recomendación basados en grandes modelos de lenguaje (LLMs) bajo arquitecturas RAG han ganado una relevancia creciente en plataformas digitales, si bien introducen nuevas fuentes de sesgo estructural que pueden comprometer su fiabilidad. En particular, el sesgo posicional, entendido como la dependencia de la recomendación respecto al orden en que los productos candidatos se presentan en el *prompt*, carece hasta la fecha de un marco estadístico riguroso para su detección en entornos de recomendación, donde la ausencia de etiquetas de relevancia limita la aplicabilidad de los métodos existentes. En este trabajo proponemos un marco formal basado en permutaciones que demuestra que la ausencia de sesgo posicional implica la uniformidad de la distribución de probabilidad sobre las posiciones del *prompt*, y empleamos esta propiedad para construir un test estadístico de bondad de ajuste que no requiere etiquetas de relevancia. La evaluación empírica sobre ocho modelos de última generación pertenecientes a las familias LLaMA, GPT-OSS y DeepSeek R1 muestra que cuatro de ellos presentan sesgo posicional estadísticamente significativo, con patrones cualitativos diferenciados entre familias, lo que pone de manifiesto que el sesgo posicional es un fenómeno real y medible en estos sistemas y subraya la necesidad de incorporar su detección en los procesos de auditoría de sistemas de recomendación basados en LLMs.

Palabras clave: Sesgo posicional · Grandes modelos de lenguaje · Sistemas de recomendación

1. Introducción

Los sistemas de recomendación constituyen un componente fundamental de las plataformas digitales modernas, con aplicaciones en ámbitos tan diversos como

el comercio electrónico, el entretenimiento o la recuperación de información. Con la aparición de los grandes modelos de lenguaje (LLMs), estos sistemas han experimentado una transformación significativa gracias a su capacidad para razonar sobre texto en lenguaje natural [11]. Una arquitectura habitual para implementar este tipo de sistemas es la generación aumentada por recuperación (RAG), en la que un módulo de recuperación selecciona un conjunto de productos candidatos y un LLM genera la recomendación final [19]. En este trabajo nos centramos en el caso de la recomendación Top-1, en el que el sistema selecciona un único producto de entre los candidatos.

Sin embargo, el uso de LLMs como núcleo de decisión introduce nuevas fuentes de sesgo. Distinguimos entre sesgos injustos, que afectan a grupos de usuarios definidos por atributos sensibles [5], y sesgos estructurales, que emergen de la propia formulación del sistema. En este trabajo nos centramos en el *sesgo posicional*, un tipo de sesgo estructural que se manifiesta cuando la probabilidad de recomendar un producto depende de la posición que ocupa en el *prompt* en lugar de su relevancia respecto a la consulta del usuario.

Aunque el sesgo posicional ha sido estudiado en dominios relacionados como las preguntas de elección múltiple [17,23] o la evaluación automática [24], los métodos existentes presentan limitaciones importantes: la mayoría requieren conjuntos de datos etiquetados con una respuesta correcta [9], condición que no se satisface en entornos de recomendación, y algunos asumen la uniformidad de la distribución de posiciones bajo la ausencia de sesgo sin una justificación formal [13]. Por tanto, las contribuciones de este trabajo son las siguientes:

- Proponemos un marco estadístico formal para la detección y análisis de sesgo posicional en sistemas de recomendación Top-1 basados en LLMs, que no requiere etiquetas de relevancia e incluye una demostración formal de que la ausencia de sesgo posicional implica la uniformidad de la distribución de probabilidad sobre las posiciones del *prompt*.
- Realizamos una evaluación empírica sobre ocho modelos de última generación pertenecientes a las familias LLaMA, GPT-OSS y DeepSeek R1, empleando un caso de uso de recomendación de productos construido a partir del conjunto de datos *Amazon ML Challenge 2023*.

Los resultados muestran que cuatro de los ocho modelos evaluados presentan sesgo posicional significativo estadísticamente, con patrones diferenciados entre familias de modelos, subrayando la importancia de incorporar su detección en los procesos de auditoría de estos sistemas. El resto del artículo se organiza como sigue: la Sección 2 revisa los trabajos relacionados, la Sección 3 presenta el marco formal y la metodología propuesta, la Sección 4 describe la configuración experimental y analiza los resultados, y la Sección 5 resume las conclusiones y las direcciones de trabajo futuro.

2. Trabajo relacionado

Los sistemas de recomendación conversacionales son aquellos que permiten al usuario interactuar con el sistema en lenguaje natural, donde la tarea del sis-

tema es entender las preferencias del mismo para así poder recomendar uno o más de los productos ofertados en la plataforma donde se implemente. La implementación de dichos sistemas se ha desarrollado usando diferentes alternativas, donde los métodos más clásicos emplean sistemas basados en reglas o enfoques basados en aprendizaje automático. Con el crecimiento de los grandes modelos de lenguaje (LLMs) y sus capacidades de generalizar a tareas para las que no han sido diseñados, los sistemas conversacionales se pueden implementar usando estos modelos como núcleo de la generación de la recomendación [11]. Un ejemplo de diseño sería usando metodologías basadas en RAG (Generación Aumentada por Recuperación), cuyo sistema se descompone en dos componentes principales: recuperación de elementos relacionados en base a una petición y generación de contenido en base a lo recuperado [19]. La arquitectura basada en RAG se puede aplicar directamente a un sistema de recomendación, donde la base de datos se compone de los productos que componen al catálogo. Dicha recomendación puede dar como resultado uno o más productos, los cuales se usarán para proveer al usuario de una experiencia personalizada.

La implementación de sistemas de recomendación usando modelos de caja negra en vez de técnicas más interpretables implica que se desconocen las reglas de decisión por las que un modelo realiza la recomendación. Esto conlleva, entre otras cosas, la potencial aparición de sesgos difíciles de detectar y mitigar, por lo que es necesario entender los potenciales sesgos que puede inducir un modelo de estas características. Dicho entendimiento primero parte de la diferenciación de los diferentes sesgos, entre los que encontramos dos grandes grupos de sesgos y a los que denominaremos como sesgos injustos y sesgos estructurales, siendo los sesgos injustos aquellos que comprometen a grupos protegidos como lo son la raza o el sexo [5,15], y los sesgos estructurales como todos aquellos que dependen de la estructura del texto que se le provee al modelo de lenguaje [5].

En este trabajo nos centraremos en el estudio del sesgo posicional, un tipo de sesgo estructural que condiciona las reglas de decisión del modelo según la posición en la que se le proveen los productos dentro del *prompt*. En términos estadísticos, implica una tendencia del modelo a seleccionar determinadas posiciones con independencia de la relevancia de los productos respecto a los intereses del usuario. Este fenómeno se ha estudiado en tareas como las Preguntas-Respuestas de elección múltiple [17,23,22], los sistemas basados en RAG [12,10], la evaluación mediante modelos de lenguaje [24,18] y los sistemas de recomendación basados en LLMs [9], atribuyéndose habitualmente su origen a los propios datos de entrenamiento y alineamiento [4,3,20]. A diferencia de las tres primeras tareas, donde su evaluación suele contar con un conjunto etiquetado, en recomendación resulta más complicada de evaluar por la ambigüedad inherente a la relevancia de un producto, donde varios candidatos podrían ser igualmente relevantes [9].

Trabajos relacionados a este ya han estudiado la presencia del sesgo posicional en los modelos de lenguaje, incluyendo el ámbito específico de la recomendación [14,2]. Sin embargo, estos trabajos o bien requieren conocer la posición de un ítem objetivo dentro de la lista de candidatos [14], o bien se limitan a una

caracterización descriptiva del sesgo mediante correlaciones de *ranking* [2], sin ofrecer en ningún caso una definición formal de ausencia de sesgo posicional ni un test de hipótesis que determine su significación estadística. A esto se suman métricas como RStd [23], que necesita datos etiquetados con una respuesta correcta, y otras basadas en la divergencia KL frente a la distribución uniforme, asumida sin justificación formal [13]. En base a estas observaciones, en este trabajo identificamos que el estudio del sesgo posicional necesita de un fundamento estadístico sólido que permita realizar test de hipótesis.

3. Formulación y metodología

En esta sección formalizamos el problema del sesgo posicional en sistemas de recomendación basados en LLMs y presentamos la metodología estadística propuesta para su detección. En primer lugar, la Sección 3.1 introduce la arquitectura RAG como marco de referencia y establece la notación probabilística necesaria para modelar el comportamiento del sistema. A continuación, la Sección 3.2 describe el test estadístico empleado para detectar el sesgo posicional, así como la matriz de inspección del sesgo como herramienta de análisis cualitativo. Finalmente, la Sección 3.3 demuestra formalmente que, en ausencia de sesgo posicional, la distribución de probabilidad sobre las posiciones del *prompt* es necesariamente uniforme, justificando así el fundamento teórico de la metodología propuesta.

3.1. Definiendo un sistema de recomendación basado en LLMs

Un sistema de recomendación basado en LLMs puede implementarse siguiendo una arquitectura RAG, la cual se descompone en dos etapas principales. En primer lugar, un módulo de recuperación selecciona los k productos más relevantes de un catálogo en base a la similitud semántica con la consulta del usuario, empleando para ello un modelo de *embeddings*. En segundo lugar, los productos recuperados se incorporan junto con la consulta del usuario en un *prompt*, que es proporcionado al LLM para que genere la recomendación final. Esta arquitectura permite al LLM razonar sobre un conjunto acotado de candidatos expresados en lenguaje natural, sin necesidad de acceder directamente al catálogo completo.

Definimos $\mathcal{D}$ como una distribución de probabilidad sobre las posibles consultas de usuario del sistema de recomendación y sus k productos asociados, de forma que muestreamos de $\mathcal{D}$ tuplas de la forma $(Q, P_1, P_2, \ldots, P_k)$, donde Q es una consulta de usuario y $P_1, P_2, \ldots, P_k$ son los k productos más relevantes seleccionados mediante la arquitectura RAG para dicha consulta. Siendo S_k el conjunto de todas las permutaciones de $\{1, 2, \ldots, k\}$ y para una instancia $X \sim \mathcal{D}$ y una permutación $\sigma \in S_k$, construimos el *prompt* del sistema de recomendación presentando los productos en el orden $P_{\sigma(1)}, P_{\sigma(2)}, \ldots, P_{\sigma(k)}$, obteniendo como resultado una recomendación por parte del LLM. Denotamos por $R(X, \sigma)$ la posición en el *prompt* del producto recomendado. Por ejemplo, si $R(X, \sigma) = 1$, el

LLM recomienda el primer producto del *prompt*, que corresponde a $P_{\sigma(1)}$ en la lista original.

En nuestro análisis, la permutación σ se muestrea de forma uniforme sobre S_k, denotado $\sigma \sim U(S_k)$. El uso de permutaciones aleatorias permite evaluar el sesgo posicional de la recomendación R, en línea con los principios del contraste basado en permutaciones [6]. Bajo esta formulación, definimos la distribución de probabilidad marginal sobre las posiciones del *prompt* como

$$p(i) = \Pr_{X \sim \mathcal{D},\, \sigma \sim U(S_k)}[R(X, \sigma) = i], \quad i \in \{1, 2, \ldots, k\}, \tag{1}$$

donde evidentemente se cumple $\sum_{i=1}^{k} p(i) = 1$. La distribución p captura la tendencia global del modelo a recomendar el producto situado en cada posición del *prompt*, promediada sobre todas las consultas posibles y todas las permutaciones. En ausencia de sesgo posicional, esta distribución debería ser uniforme: $p(1) = p(2) = \cdots = p(k) = 1/k$, ya que ninguna posición del *prompt* debería ser sistemáticamente preferida con independencia del contenido semántico de los productos. Esta propiedad se demuestra formalmente en la Sección 3.3.

3.2. Metodología para la detección de sesgo posicional

Para detectar el sesgo posicional, proponemos contrastar la hipótesis nula de que la distribución p definida en la Ecuación 1 es uniforme. Como se demuestra en la Sección 3.3, esta hipótesis es equivalente a la ausencia de sesgo posicional. Para ello, empleamos el contraste de bondad de ajuste Chi-cuadrado sobre muestras de p, el cual requiere un número mínimo de $k \cdot 30$ muestras para garantizar la validez asintótica del test.

Como alternativa más directa, existe un contraste basado en la matriz de contingencia de las variables Y y σ, donde $Y = \sigma(R(X, \sigma))$ representa el índice original del producto recomendado. Bajo la ausencia de sesgo posicional, estas dos variables deberían ser estadísticamente independientes, lo que puede contrastarse mediante el test Chi-cuadrado de independencia sobre dicha matriz. Si bien este enfoque es más directo comparado con el test de uniformidad, el medir la independencia con dicha matriz requiere un número significativamente mayor de muestras para garantizar que cada celda de la matriz de contingencia contenga un mínimo de 5 observaciones, condición necesaria para la validez asintótica del test.

Más allá de la detección estadística, resulta de interés caracterizar cualitativamente el patrón de sesgo posicional observado. Con este fin, introducimos la *matriz de inspección del sesgo*, donde cada fila i representa un producto según su posición original en la lista ordenada devuelta por el módulo de recuperación, y cada columna j representa una posición del *prompt* tras aplicar la permutación. La celda (i, j) recoge la media del número de veces que el producto i es recomendado cuando la permutación lo sitúa en la posición j del *prompt*, esto es,

$$A_{i,j} = \frac{1}{(k-1)!} \sum_{\sigma \in S_k \,:\, \sigma(i) = j} \left(\sum_{X \sim \mathcal{D}} \mathbf{1}[R(X, \sigma) = j] \right). \tag{2}$$

Esta matriz permite identificar qué posiciones del *prompt* son sistemáticamente favorecidas o penalizadas por el modelo con independencia del producto concreto que las ocupe. Interpretada por columnas, una columna j con valores globalmente elevados indica una preferencia posicional positiva hacia la posición j, mientras que valores globalmente reducidos revelan una tendencia negativa. Esta visualización complementa el contraste estadístico definido, ofreciendo una descripción cualitativa de la estructura del sesgo detectado.

3.3. Uniformidad bajo la ausencia de sesgo posicional

En esta sección demostramos formalmente que, en ausencia de sesgo posicional, la distribución p definida en la Ecuación 1 es uniforme. Comenzamos formalizando la ausencia de sesgo posicional.

Definición 1. *Decimos que una función de recomendación R está* libre de sesgo posicional *respecto a una distribución $\mathcal{D}$ si, para casi toda instancia $X \sim \mathcal{D}$, existe una función $g(X) \in \{1, \dots, k\}$ tal que el producto recomendado es el mismo para cualquier permutación, esto es,*

$$\sigma(R(X, \sigma)) = g(X) \quad \forall \sigma \in S_k.$$

Esta definición expresa que, en ausencia de sesgo posicional, la recomendación del modelo debe depender únicamente del contenido semántico de los productos y de la consulta del usuario, y no del orden en que los productos se presentan en el *prompt*.

Proposición 1. *En ausencia de sesgo posicional en el sentido de la Definición 1, se tiene que $p(i) = \frac{1}{k}$ para todo $i \in \{1, 2, \dots, k\}$.*

Demostración. Fijamos una instancia $X = x$. Por la Definición 1, existe un índice $g(x) \in \{1, \dots, k\}$ tal que $\sigma(R(x, \sigma)) = g(x)$ para toda permutación $\sigma \in S_k$, lo que equivale a que $R(x, \sigma) = \sigma^{-1}(g(x))$. Por tanto, para cualquier posición $i \in \{1, \dots, k\}$, tenemos que

$$\begin{aligned} p(i \mid X = x) &= \mathbb{E}_{\sigma \sim U(S_k)} \mathbf{1}[R(x, \sigma) = i] \\ &= \mathbb{E}_{\sigma \sim U(S_k)} \mathbf{1}\left[\sigma^{-1}(g(x)) = i\right] \\ &= \mathbb{E}_{\sigma \sim U(S_k)} \mathbf{1}[g(x) = \sigma(i)] \\ &= \frac{1}{k}, \end{aligned}$$

donde la última igualdad se debe a que, al promediar sobre todas las permutaciones uniformes, $\sigma(i)$ toma cada valor de $\{1, \dots, k\}$ con la misma probabilidad $\frac{1}{k}$, con independencia del valor fijo $g(x)$. Marginalizando finalmente sobre $X \sim \mathcal{D}$ obtenemos $p(i) = \mathbb{E}_{X \sim \mathcal{D}}[p(i \mid X)] = \frac{1}{k}$ para todo $i \in \{1, \dots, k\}$. □

La Proposición 1 justifica el uso del contraste de bondad de ajuste Chi-cuadrado frente a la distribución uniforme como test estadístico para la detección

de sesgo posicional: si el sistema está libre de sesgo, p es necesariamente uniforme, por lo que una desviación significativa detectada por el contraste constituye evidencia de sesgo posicional. El recíproco no se sostiene necesariamente, pues un modelo sesgado podría, en casos particulares, dar lugar a una distribución agregada uniforme, por lo que el no rechazo de la hipótesis nula no garantiza la ausencia de sesgo, sino únicamente la falta de evidencia frente a esta forma de sesgo.

4. Experimentación y resultados

En esta sección presentamos la evaluación empírica del sesgo posicional en sistemas de recomendación basados en LLMs, siguiendo la metodología descrita en la Sección 3.2. En primer lugar, la Sección 4.1 describe la configuración experimental, incluyendo la construcción del conjunto de datos de evaluación y los detalles de implementación del sistema. A continuación, la Sección 4.2 presenta los resultados obtenidos, organizados en dos bloques: la aplicación del test estadístico de uniformidad y el análisis cualitativo del sesgo mediante la matriz de inspección.

4.1. Configuración experimental

El conjunto de datos de evaluación se construye a partir de dos fuentes principales. Por un lado, se generan 720 consultas de usuario sintéticas mediante plantillas que cubren distintas categorías de productos y preferencias de usuario. Por otro lado, el catálogo de productos se obtiene del conjunto de datos *Amazon ML Challenge 2023*[3], del que se muestrean 50,000 productos, cada uno de ellos con un título y una descripción. A partir de estas dos fuentes, se construye el conjunto de datos RAG aplicando el esquema de recuperación descrito en la Sección 3.1, empleando el modelo de *embeddings* BAAI [21] para calcular la similitud semántica entre las consultas y los títulos de los productos y seleccionando los $k = 5$ más similares para cada consulta. Esto resulta en 720 tuplas de la forma $(Q, P_1, P_2, P_3, P_4, P_5)$. Para cada una de estas tuplas, se consideran las 120 permutaciones posibles de los 5 productos candidatos, lo que resulta en un total de $720 \cdot 120 = 86{,}400$ instancias de evaluación.

Los modelos evaluados en este trabajo pertenecen a cuatro familias de LLMs de última generación. En primer lugar, la familia LLaMA 3.1, representada por sus variantes de 8B y 70B parámetros [7]. En segundo lugar, LLaMA 4 Scout [1]. En tercer lugar, la familia GPT-OSS, con sus variantes de 20B y 120B parámetros [16]. Finalmente, la familia DeepSeek R1, representada por tres variantes: la destilación sobre LLaMA de 8B parámetros y las destilaciones sobre Qwen de 7B y 32B parámetros [8]. Todos los modelos se evalúan con temperatura cero, garantizando así el determinismo de las recomendaciones y permitiendo aislar el efecto del orden de los productos de cualquier variabilidad estocástica introducida por el proceso de generación.

[3] Disponible en
https://www.kaggle.com/datasets/ashisparida/amazon-ml-challenge-2023

Tabla 1. Resultados del test de bondad de ajuste Chi-cuadrado aplicado sobre la distribución empírica $p(i)$ para cada modelo evaluado. El p-valor se estima como la mediana de 500 iteraciones de *bootstrap*, seleccionando en cada una de ellas una permutación por consulta. Los valores en negrita indican un α inferior a 0,05.

Model name	p-valor
DeepSeek R1 32B (Qwen)	0.366757
DeepSeek R1 7B (Qwen)	**0.000080**
DeepSeek R1 8B (LLaMA)	0.359945
LLaMA 3.1 8B	0.305323
LLaMA 4 Scout	0.250782
GPT-OSS 120B	**0.029484**
GPT-OSS 20B	**0.000181**
LLaMA 3.1 70B	**0.002477**

4.2. Resultados y análisis

Aplicación del test estadístico Para evaluar la presencia de sesgo posicional, aplicamos la metodología descrita en la Sección 3.3 sobre las 86,400 instancias del conjunto de evaluación. Dado que el test de bondad de ajuste Chi-cuadrado requiere muestras independientes de p, no podemos emplear directamente todas las permutaciones de cada consulta de forma simultánea. En su lugar, estimamos el p-valor mediante un procedimiento de *bootstrap*: en cada iteración, se selecciona aleatoriamente una permutación por consulta, obteniendo 720 muestras independientes de p, y se aplica el test Chi-cuadrado. Este procedimiento se repite 500 veces, reportando la mediana de los p-valores como estimación final, robusta frente a la variabilidad del muestreo aleatorio de permutaciones. Los resultados se recogen en la Tabla 1.

A partir de la Tabla 1, se observa que 4 de los 8 modelos evaluados son detectados como posicionalmente sesgados bajo el umbral de significación estándar $\alpha = 0{,}05$. Cabe destacar que este resultado se obtiene a partir de un conjunto de tan solo 720 consultas con sus respectivas permutaciones, lo que pone de manifiesto la eficiencia en términos de muestras de la metodología propuesta. Los modelos detectados como posicionalmente sesgados son DeepSeek R1 7B (Qwen), GPT-OSS 120B, GPT-OSS 20B y LLaMA 3.1 70B, mientras que los cuatro modelos restantes no presentan evidencia estadística suficiente para rechazar la hipótesis nula de uniformidad.

La Figura 1 muestra la distribución empírica de probabilidad $p(i)$ para cada modelo evaluado, permitiendo visualizar cualitativamente los patrones de sesgo posicional detectados. Se observa que los modelos de la familia GPT-OSS y LLaMA 3.1 70B muestran una clara tendencia a seleccionar productos situados en las primeras posiciones del *prompt*. El modelo DeepSeek R1 7B (Qwen) presenta una preferencia marcada por la segunda posición, mientras que los modelos no detectados como sesgados muestran distribuciones más próximas a la uniformidad.

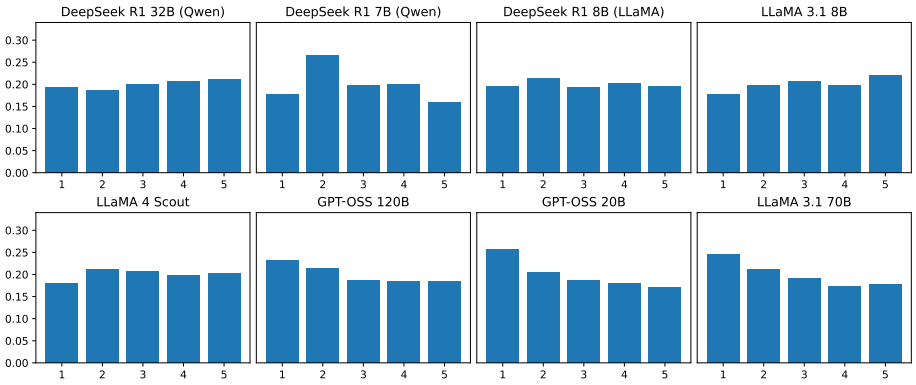


Figura 1. Distribución empírica $p(i)$ para cada modelo evaluado, donde cada barra indica la probabilidad de recomendar el producto en la posición i del *prompt*. En ausencia de sesgo posicional, todas las barras deberían tener altura $1/k$.

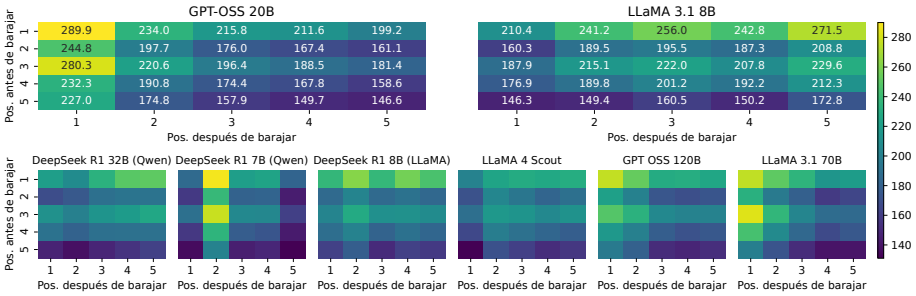


Figura 2. Matriz de inspección del sesgo posicional definida en la Ecuación 2 para cada modelo evaluado. Cada fila i representa un producto según su posición original en el ranking del módulo de recuperación, y cada columna j representa una posición del *prompt* tras aplicar la permutación.

En conjunto, estos resultados confirman que el sesgo posicional es un fenómeno estadísticamente detectable en varios modelos de última generación, cuya dirección y magnitud varían considerablemente entre familias, lo que sugiere un papel relevante de las diferencias arquitectónicas o de entrenamiento.

Inspección del sesgo posicional Para caracterizar cualitativamente la estructura del sesgo posicional detectado, analizamos la matriz de inspección del sesgo descrita en la Sección 3.2, cuyos resultados se muestran en la Figura 2 para todos los modelos evaluados.

Los patrones observados en la Figura 2 son consistentes con los descritos en el apartado anterior. Los modelos de la familia GPT-OSS muestran una clara preferencia por los productos situados en las primeras posiciones del *prompt*

con independencia del producto concreto que las ocupe, mientras que LLaMA 3.1 8B, pese a no presentar un sesgo posicional estadísticamente detectable, exhibe el patrón contrario, favoreciendo sistemáticamente las últimas posiciones. Estos resultados son coherentes con observaciones previas sobre cómo los LLMs atienden a diferentes regiones del *prompt*, donde la información situada al inicio o al final suele tener mayor influencia en la decisión del modelo [12].

Adicionalmente, se observa que los productos situados en las primeras posiciones antes de aplicar la permutación, es decir, los productos mejor valorados por el módulo de recuperación, tienden a ser seleccionados con mayor frecuencia con independencia de la posición a la que sean desplazados por la permutación. Este comportamiento es esperable, dado que el módulo de recuperación ordena los productos por similitud semántica con la consulta, por lo que los productos más relevantes son también los más frecuentemente recomendados por el LLM.

En definitiva, la matriz de inspección del sesgo confirma y enriquece los resultados obtenidos mediante el test estadístico, permitiendo identificar no solo la presencia de sesgo posicional sino también su dirección y estructura, lo que constituye información valiosa de cara al diseño y auditoría de sistemas de recomendación basados en LLMs.

5. Conclusiones y trabajo futuro

En este trabajo se ha propuesto un marco estadístico para detectar y analizar el sesgo posicional en sistemas de recomendación basados en LLMs bajo arquitecturas RAG. La propuesta se basa en permutar aleatoriamente los productos candidatos dentro del prompt y estudiar la distribución de las posiciones seleccionadas por el modelo. A partir de una formalización probabilística, se demuestra que, en ausencia de sesgo posicional, dicha distribución debe ser uniforme, lo que justifica el uso de un test de bondad de ajuste Chi-cuadrado como mecanismo de detección sin necesidad de etiquetas externas de relevancia.

La evaluación empírica sobre ocho modelos de las familias LLaMA, GPT-OSS y DeepSeek R1 muestra que cuatro presentan sesgo posicional estadísticamente significativo. La matriz de inspección propuesta revela además patrones diferenciados entre modelos, desde una clara preferencia por las primeras posiciones hasta distribuciones próximas a la uniformidad. Estos resultados evidencian que el sesgo posicional es un fenómeno real, medible y dependiente del modelo, por lo que debería considerarse en auditorías de sistemas de recomendación basados en LLMs.

Como limitación principal, cabe señalar que la Proposición 1 se deriva directamente de la simetría del muestreo uniforme de permutaciones; no obstante, su formalización resulta necesaria, pues trabajos previos han asumido dicha uniformidad sin justificación explícita (Sección 2). Además, el marco propuesto adopta una definición estricta de ausencia de sesgo posicional, basada en la invariancia de la recomendación frente a permutaciones, que no incorpora la relevancia semántica de los productos y no distingue completamente entre una preferencia estructural por ciertas posiciones y una regla de decisión estable asociada a di-

ferencias reales de relevancia. Como trabajo futuro, resulta necesario incorporar estimaciones de relevancia o preferencias humanas, así como evaluar el sesgo en escenarios Top-k, distintos tamaños de lista candidata y sistemas RAG en dominios reales.

Agradecimientos. Esta investigación es parte del Proyecto TSI-100927-2023-1, financiado por Plan de Recuperación, Transformación y Resiliencia Europeo, de la Unión Europea-Next Generation EU a través del Ministerio para la Transformación Digital y de la Función Pública y está parcialmente financiada por el Ministerio de Ciencia y Tecnología de España en el marco del proyecto PID2023-150070NB-I00, financiado por MICIU/AEI/10.13039/501100011033 y por FEDER, UE.

Declaración de Intereses. Los autores declaran que no tienen intereses en conflicto relevantes para el contenido de este artículo.

Referencias

1. AI, M.: Introducing llama 4: Advancing multimodal intelligence (2024), https://ai.meta.com/blog/llama-4-multimodal-intelligence/
2. Bito, E., Ren, Y., He, E.: Evaluating position bias in large language model recommendations (2025), https://arxiv.org/abs/2508.02020
3. Chen, J., Dong, H., Wang, X., et al.: Bias and Debias in Recommender System: A Survey and Future Directions. ACM Trans. Inf. Syst. **41**(3) (Feb 2023), place: New York, NY, USA Publisher: Association for Computing Machinery
4. Collins, A., Tkaczyk, D., Aizawa, A., et al.: Position Bias in Recommender Systems for Digital Libraries. In: Chowdhury, G., McLeod, J., Gillet, V., Willett, P. (eds.) Transforming Digital Worlds. pp. 335–344. Springer International Publishing, Cham (2018)
5. Gallegos, I.O., Rossi, R.A., Barrow, J., et al.: Bias and fairness in large language models: A survey. Computational Linguistics **50**(3), 1097–1179 (Sep 2024)
6. Good, P.: Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses. Springer, New York, NY, USA, 3 edn. (2013)
7. Grattafiori, A., et al.: The Llama 3 Herd of Models (Nov 2024), arXiv:2407.21783 [cs]
8. Guo, D., Yang, D., Zhang, H., et al.: Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. Nature **645**, 633–638 (2025)
9. Hou, Y., Zhang, J., Lin, Z., et al.: Large Language Models are Zero-Shot Rankers for Recommender Systems. In: Advances in Information Retrieval: 46th European Conference on Information Retrieval, ECIR 2024, Glasgow, UK, March 24–28, 2024, Proceedings, Part II. pp. 364–381. Springer-Verlag, Berlin, Heidelberg (Mar 2024)
10. Hsieh, C.Y., Chuang, Y.S., Li, C.L., et al.: Found in the middle: Calibrating positional attention bias improves long context utilization. In: Ku, L.W., Martins, A., Srikumar, V. (eds.) Findings of the Association for Computational Linguistics: ACL 2024. pp. 14982–14995. Association for Computational Linguistics, Bangkok, Thailand (Aug 2024)
11. Li, L., Zhang, Y., Liu, D., et al.: Large language models for generative recommendation: A survey and visionary discussions. In: Calzolari, N., Kan, M.Y., Hoste,

V., et al. (eds.) Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024). pp. 10146–10159. ELRA and ICCL, Torino, Italia (May 2024)
12. Liu, N.F., Lin, K., Hewitt, J., et al.: Lost in the middle: How language models use long contexts. Transactions of the Association for Computational Linguistics **12**, 157–173 (2024)
13. Liusie, A., Fathullah, Y., Gales, M.: Teacher-student training for debiasing: General permutation debiasing for large language models. In: Ku, L.W., Martins, A., Srikumar, V. (eds.) Findings of the Association for Computational Linguistics: ACL 2024. pp. 1376–1387. Association for Computational Linguistics, Bangkok, Thailand (Aug 2024)
14. Ma, T., Cheng, Y., Zheng, Z., Zhu, H., Xiong, H.: Large language models are not stable recommender systems: A position bias perspective. In: Knowledge Science, Engineering and Management: 18th International Conference, KSEM 2025, Macao, China, August 4–7, 2025, Proceedings, Part I. p. 415–429. Springer-Verlag, Berlin, Heidelberg (2025)
15. Navigli, R., Conia, S., Ross, B.: Biases in large language models: origins, inventory, and discussion. ACM Journal of Data and Information Quality **15**(2), 1–21 (2023)
16. OpenAI: gpt-oss-120b & gpt-oss-20b model card (2025)
17. Pezeshkpour, P., Hruschka, E.: Large language models sensitivity to the order of options in multiple-choice questions. In: Duh, K., Gomez, H., Bethard, S. (eds.) Findings of the Association for Computational Linguistics: NAACL 2024. pp. 2006–2017. Association for Computational Linguistics, Mexico City, Mexico (Jun 2024)
18. Shi, L., Ma, C., Liang, W., et al.: Judging the judges: A systematic study of position bias in LLM-as-a-judge. In: Inui, K., Sakti, S., Wang, H., et al. (eds.) Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics. pp. 292–314. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics, Mumbai, India (Dec 2025)
19. Wang, L., Lim, E.P.: Zero-Shot Next-Item Recommendation using Large Pretrained Language Models (Apr 2023), arXiv:2304.03153 [cs]
20. Wu, L., Zheng, Z., Qiu, Z., et al.: A survey on large language models for recommendation. World Wide Web **27**(5), 60 (2024)
21. Xiao, S., Liu, Z., Zhang, P., et al.: C-Pack: Packed Resources For General Chinese Embeddings. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 641–649. SIGIR '24, Association for Computing Machinery, New York, NY, USA (Jul 2024)
22. Yang, Z., Jian, P., Li, C.: Option symbol matters: Investigating and mitigating multiple-choice option symbol bias of large language models. In: Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers). pp. 1902–1917. Association for Computational Linguistics, Albuquerque, New Mexico (Apr 2025)
23. Zheng, C., Zhou, H., Meng, F., et al.: Large Language Models Are Not Robust Multiple Choice Selectors (2024)
24. Zheng, L., Chiang, W.L., Sheng, Y., et al.: Judging llm-as-a-judge with mt-bench and chatbot arena. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. NIPS '23, Curran Associates Inc., Red Hook, NY, USA (2023)