COLECCIÓN

MITOS Y REALIDADES DE LA IA

Entre la promesa y la realidad

PROBABILIDAD, ALEATORIEDAD Y DETERMINISMO EN LAS RESPUESTAS DE UNA IA

Número 11



MRCI

anejo de ecursos y ontroles nteligentes

DR. JOSÉ ANTONIO PÉREZ RAMOS

PROBABILIDAD, ALEATORIEDAD Y DETERMINISMO EN LAS RESPUESTAS DE UNA IA

SOBRE EL AUTOR

Doctor en Ciencias de lo Fiscal por el Instituto de Especialización para Ejecutivos. Maestro en Fiscal por la Universidad Autónoma "Benito Juárez" de Oaxaca. Maestro en Estudios Avanzados en Derechos Humanos por la Global Open University. Licenciado en Contaduría Pública por la Universidad Autónoma "Benito Juárez" de Oaxaca y Licenciado en Filosofía por la Universidad Nacionalista de México.

Socio fundador y director general de la Firma Manejo de Recursos y Controles Inteligentes (Firma MRCI). Fiscalista del año 2009 por la revista *Defensa Fiscal*. Doctor honoris causa por la Asamblea General del Claustro Doctoral Iberoamericano (2016). Doctor honoris causa por el Instituto Mexicano de Líderes de Excelencia (2017). Doctor honoris causa por 1Million Startups, Latinomics, Leaderships Forum y la Fundación Humanist World (2018). Doctor honoris causa por el Claustro de Doctores de la AISOL (American International School of Law) (2019).

Autor de diversas obras, entre las que destacan *Ius mutantur o del derecho que cambia* (Tirant Lo Blanch, 2025), *Futurología* (MRCI, 2025) y *Hasta que la muerte nos separe* (MRCI, 2025). Ha participado como coordinador en *Cuestiones tributarias. Problemas y controversias en el México actual* (Tirant Lo Blanch, 2023), *El costo de la justicia* (Apex Iuris, 2019), *Remuneraciones estratégicas inteligentes* (MRCI, 2015) y *Retos de la justicia tributaria* (Tirant Lo Blanch-AMDF, 2025).

Titular de *Pie de Página*, programa para la divulgación de los derechos humanos en México, a través de la plataforma Legal Cloud. Expositor en foros nacionales e internacionales. Colaborador en diversas revistas jurídicas, entre las que destacan *Defensa Fiscal* y *Legal Cloud*.

PROBABILIDAD, ALEATORIEDAD Y DETERMINISMO EN LAS RESPUESTAS DE UNA IA

DR. JOSÉ ANTONIO PÉREZ RAMOS

PRIMERA EDICIÓN, ATEMPORAL

Derechos reservados, propiedad de
Dr. José Antonio Pérez Ramos

Comentarios y opiniones: investigacion@mrci.com.mx

Título original: Probabilidad, aleatoriedad, y determinismo en las respuestas de una IA.

Colección mitos y realidades de la inteligencia artificial.

Entre la promesa y la realidad.

Autor: Dr. José Antonio Pérez Ramos.

Queda prohibida la reproducción total y parcial de esta obra denominada; PROBABILIDAD, ALEATORIEDAD, Y DETERMINISMO EN LAS RESPUESTAS DE UNA IA, por cualquier medio, sin autorización escrita del autor.

PRINTED IN MEXICO
IMPRESO EN MÉXICO

ÍNDICE

EL PROBLEMA DE ATRIBUIR TODO CAMBIO AL AZAR ..... - 7 -
MÉTODO Y LÍMITES DE LA REVISIÓN ..... - 9 -
QUÉ REPRESENTA UNA PROBABILIDAD EN UN MODELO DE LENGUAJE ..... - 10 -
UN VOCABULARIO PEQUEÑO PARA EVITAR EQUÍVOCOS ..... - 12 -
UNA DISTRIBUCIÓN NO OBLIGA A SORTEAR ..... - 15 -
MUESTREO CON PESOS DESIGUALES ..... - 17 -
LA COINCIDENCIA TAMBIÉN PUEDE SER ALEATORIA ..... - 18 -
ELEGIR EL SIGUIENTE PASO NO EQUIVALE A ELEGIR TODA LA RESPUESTA ..... - 20 -
LA LONGITUD Y LA PROBABILIDAD CONJUNTA ..... - 22 -
SEMILLAS Y SECUENCIAS SEUDOALEATORIAS ..... - 24 -
UN GENERADOR DIDÁCTICO COMPLETAMENTE VISIBLE ..... - 25 -
LA EJECUCIÓN NUMÉRICA TAMBIÉN FORMA PARTE DEL RESULTADO ..... - 29 -
CUÁNDO UNA DIFERENCIA PEQUEÑA CAMBIA LA SELECCIÓN ..... - 31 -
DE UNA DIFERENCIA LOCAL A UNA TRAYECTORIA DISTINTA ..... - 33 -
VERDAD Y PROBABILIDAD DE EMISIÓN ..... - 34 -
EL ALCANCE DE LA EVIDENCIA SOBRE RESPUESTAS FALSAS ..... - 35 -
CONFIANZA VERBAL Y CALIBRACIÓN ..... - 37 -
COINCIDENCIA ENTRE RESPUESTAS Y ACUERDO CON LA EVIDENCIA ..... - 38 -
CÓMO INVESTIGAR DOS RESPUESTAS DIFERENTES ..... - 39 -
CASO DE UNA RESPUESTA QUE CAMBIA DE REDACCIÓN ..... - 40 -
CASO DE UNA CONTRADICCIÓN QUE EXIGE UNA FUENTE ..... - 41 -
CASO DE IGUALDAD QUE OCULTA UNA DEBILIDAD ..... - 43 -
UN REGISTRO MÍNIMO PARA DISTINGUIR CAUSAS ..... - 44 -
MITOS Y OBJECIONES QUE REQUIEREN UNA RESPUESTA PRECISA ..... - 46 -
LA OBJECIÓN DE QUE TODO PROGRAMA ES DETERMINISTA ..... - 47 -
LA OBJECIÓN DE QUE QUITAR EL AZAR ELIMINA EL ERROR ..... - 48 -
LO QUE AÚN NO PUEDE CONCLUIRSE ..... - 49 -
IMPLICACIONES PARA UTILIZAR Y EVALUAR UNA IA ..... - 50 -
CONCLUSIÓN ..... - 53 -
REFERENCIAS ..... - 54 -



El problema de atribuir todo cambio al azar

Una persona formula dos veces una pregunta a una aplicación de inteligencia artificial y recibe respuestas distintas; si ambas explican correctamente el asunto con palabras diferentes, la variación puede resultar útil, pero si cambian una cantidad, una fecha o una conclusión, aparece un problema que exige investigación y que no queda resuelto con la frase «la IA funciona con probabilidades».

Esa frase reúne cuestiones que conviene separar desde el principio: cómo se representan las alternativas, cómo se elige una de ellas y cómo se comprueba lo que se afirma; una distribución numérica puede calcularse mediante operaciones deterministas, una selección puede incorporar muestreo y el texto elegido puede ser correcto o incorrecto con independencia de que vuelva a aparecer en la siguiente ejecución.

La pregunta central de este artículo es qué permite afirmar, y qué no permite afirmar, el carácter probabilístico de una respuesta; la tesis es que la probabilidad describe una asignación de peso a resultados posibles bajo condiciones

especificadas, mientras que la aleatoriedad describe un mecanismo o modelo de variación y el determinismo describe una relación entre estados completos de entrada y salida.

La repetibilidad requiere una precisión adicional, porque suele observarse desde fuera: una persona puede repetir la misma frase visible sin repetir el contexto completo del sistema; la verdad, por su parte, exige un criterio de contraste que no se obtiene contando cuántas veces aparece una cadena de palabras.

El análisis se concentra en generación autorregresiva de texto, es decir, en modelos que construyen una secuencia condicionando cada paso a lo que ya está disponible; esta elección permite explicar mecanismos de uso frecuente sin extenderlos indebidamente a toda la IA, que también incluye clasificadores, sistemas de reglas, buscadores, controladores y combinaciones de componentes con comportamientos diferentes.

Método y límites de la revisión

Se realizó una revisión documental orientada por conceptos, con consulta de publicaciones originales sobre modelado del lenguaje, decodificación, calibración y evaluación de veracidad, además de documentación oficial sobre ejecución reproducible; las fuentes se seleccionaron por su capacidad para sostener una distinción concreta, sin pretensión de revisión sistemática exhaustiva ni de evaluación experimental de una aplicación comercial.

Las formulaciones matemáticas y los casos de trabajo son construcciones explicativas, sus valores no proceden de medir un modelo real; cuando una operación se presenta como cálculo, sus supuestos se hacen explícitos y permiten reproducirla con aritmética elemental, mientras que los resultados de investigaciones publicadas se conservan vinculados a su fecha, tarea y alcance.

La documentación técnica se consultó el 3 de octubre de 2026 y se identifica por la versión visible; una documentación de biblioteca informa sobre esa biblioteca, pero no permite asegurar qué biblioteca, versión o configuración utiliza por dentro un servicio determinado,



incluso cuando su interfaz presenta controles con nombres semejantes.

El objetivo tampoco es describir aquí todas las técnicas de muestreo ni todas las causas de información falsa; temperatura, restricciones del conjunto de candidatos y otros ajustes merecen un examen operativo propio, mientras que este artículo establece las distinciones necesarias para comprender qué cambia realmente cuando se aplican esos procedimientos.

Qué representa una probabilidad en un modelo de lenguaje

Una probabilidad siempre necesita un objeto: puede referirse a un token, a una secuencia, a una etiqueta, a un resultado de una prueba o a la corrección de una respuesta; escribir un número entre cero y uno sin identificar ese objeto permite confundir magnitudes que tienen el mismo formato, pero no el mismo significado.

Llamaremos contexto a la información que recibe el modelo en un paso de generación y token a una unidad de la secuencia representada por el sistema; para mantener

el razonamiento legible, los ejemplos usarán letras o palabras como si fueran unidades completas, aunque una implementación real puede dividir una palabra en varios tokens.

Bengio et al. (2003) presentan el modelado estadístico del lenguaje mediante probabilidades condicionales de la siguiente palabra y expresan la probabilidad de una secuencia como producto de esas probabilidades; su trabajo ofrece un antecedente primario de la distinción entre estimar una distribución y seleccionar una continuación, sin constituir una descripción de todos los modelos actuales.

La expresión p(A | C) se lee «probabilidad asignada a A dado el contexto C»; la barra vertical señala una condición, de modo que p(A | C) y p(A | D) pueden ser diferentes sin contradicción si C y D contienen información distinta, del mismo modo que una previsión condicionada a lluvia no tiene por qué coincidir con una previsión condicionada a cielo despejado.

Esta notación tampoco establece por sí misma que el número sea la frecuencia verdadera del mundo; p puede



designar la distribución estimada por un modelo y, en ese caso, su interpretación depende de la tarea con la que fue construida y evaluada, no de la apariencia decimal de la salida.

Un vocabulario pequeño para evitar equívocos

Consideremos un ejemplo hipotético con tres continuaciones posibles, A, B y C, a las que un modelo asigna respectivamente 0.50, 0.30 y 0.20 bajo un contexto fijo; los valores son no negativos y suman uno, por lo que forman una distribución discreta válida sobre esas tres alternativas.

A tiene mayor probabilidad individual que B o C, pero las alternativas diferentes de A reúnen en conjunto 0.50; por ello, «la opción más probable» no equivale a «una opción casi segura», ni siquiera a una opción cuya probabilidad supere la suma de todas las demás.

La misma observación se vuelve importante cuando existen muchas alternativas: una opción puede encabezar una lista con una probabilidad modesta y quedar muy por debajo de uno; el rango que ocupa informa de una

comparación con sus competidoras, mientras que su probabilidad informa de cuánta masa le corresponde en el conjunto definido.

La tabla 1 organiza preguntas que suelen aparecer mezcladas en una conversación sobre IA; cada fila exige información distinta y ninguna respuesta afirmativa autoriza, por sí sola, a responder afirmativamente las demás.

Tabla 1. *Preguntas diferentes detrás de cinco términos frecuentes*

Término	**Pregunta que permite formular**	**Lo que no acredita por sí solo**
Probabilístico	¿Cómo se distribuye el peso entre resultados posibles?	Que se haga un sorteo en cada ejecución
Aleatorio	¿Qué mecanismo introduce variación según una distribución?	Que todas las alternativas tengan igual peso

Término	Pregunta que permite formular	Lo que no acredita por sí solo
Determinista	¿El mismo estado completo produce la misma salida?	Que la salida sea correcta
Repetible	¿Bajo las condiciones declaradas se obtiene el mismo resultado?	Que esas condiciones se hayan conservado en otro entorno
Verdadero	¿La afirmación coincide con los hechos o criterios pertinentes?	Que se produzca siempre la misma redacción

Nota. Elaboración propia; las definiciones se usan como distinciones operativas del artículo y requieren especificar el nivel del sistema al que se aplican.

Es útil observar la palabra «estado», que forma parte de la definición operativa de determinismo: un estado completo puede incluir datos que el usuario no escribió en su último



mensaje; por esa razón, una comparación de dos pantallas no demuestra automáticamente que el cálculo recibió exactamente lo mismo.

Una explicación probabilística tampoco reduce la generación a pegar frases vistas previamente; el modelo de Bengio et al. (2003) utiliza representaciones distribuidas para asignar probabilidad a combinaciones nuevas, lo cual basta para mostrar que modelar regularidades y repetir literalmente un registro son operaciones diferentes, sin atribuir comprensión humana a la primera.

El punto decisivo es mantener identificado el evento medido: si el sistema asigna un peso al siguiente token, interpretar ese peso como probabilidad de que todas las afirmaciones del documento sean verdaderas cambia de objeto sin justificación; esa sustitución es un error conceptual antes de ser un problema de implementación.

Una distribución no obliga a sortear

Supongamos que las probabilidades del ejemplo anterior ya están calculadas y guardadas en una tabla; todavía falta decidir qué hacer con ellas, pues el mismo conjunto de



valores permite elegir siempre A, sortear entre A, B y C, devolver la lista completa o abstenerse hasta que una condición adicional se cumpla.

La regla que escoge el mayor valor suele denominarse selección voraz o greedy cuando se aplica paso a paso; con las probabilidades 0.50, 0.30 y 0.20, devolvería A en cada ejecución ideal que conserve la distribución y la regla, sin necesidad de extraer un número aleatorio.

Esto proporciona un contraejemplo directo al razonamiento «si hay probabilidades, tiene que haber respuestas distintas»; una función puede calcular números probabilísticos de forma determinista y otra función puede seleccionar determinísticamente entre ellos, de manera que la naturaleza de la representación no determina por sí sola la naturaleza del procedimiento de elección.

Si dos opciones empatan, la definición del procedimiento debe incluir una regla de desempate; elegir la primera según un orden fijo conserva el determinismo, mientras que sortear entre las empatadas introduce un mecanismo distinto, por lo que la palabra «máximo» no describe completamente una implementación.



Muestreo con pesos desiguales

Para construir un sorteo que respete la distribución del ejemplo, puede dividirse el intervalo de cero a uno en tres partes: A ocupa desde 0 hasta antes de 0.50, B desde 0.50 hasta antes de 0.80 y C desde 0.80 hasta antes de 1; un número uniforme tomado de ese intervalo selecciona la opción correspondiente a la región donde cae.

Aunque el número auxiliar sea uniforme, la salida categórica no lo es: el intervalo de A mide 0.50, el de B mide 0.30 y el de C mide 0.20; confundir la uniformidad del número utilizado por el procedimiento con la uniformidad de los resultados borraría precisamente la información aportada por la distribución.

En diez extracciones independientes bajo esas condiciones, el número esperado de apariciones de A es cinco, el de B es tres y el de C es dos; «esperado» identifica un promedio matemático entre repeticiones hipotéticas del experimento, no una obligación de que cada grupo de diez cumpla exactamente esas cantidades.

De hecho, la probabilidad de observar exactamente cinco A, tres B y dos C en diez extracciones es $2{,}520 \times 0.50^5 \times 0.30^3 \times 0.20^2 = 0.08505$, equivalente a 8.505 %; el factor 2,520 cuenta los órdenes posibles de esas diez posiciones, por lo que el reparto que coincide con los valores esperados no es el único resultado permitido ni ocurre necesariamente en la mayoría de las tandas.

La independencia es aquí un supuesto deliberado: significa que conocer los resultados anteriores no modifica la distribución de la extracción siguiente; si el procedimiento actualiza el contexto con cada salida, ese supuesto puede dejar de describir la secuencia completa y entonces deben utilizarse probabilidades condicionadas al historial correspondiente.

La coincidencia también puede ser aleatoria

Dos extracciones independientes pueden devolver el mismo resultado, incluso cuando el mecanismo de selección es genuinamente estocástico en el modelo matemático; en el ejemplo, coincidir en A tiene probabilidad 0.25, coincidir en B tiene 0.09 y coincidir en C tiene 0.04, de modo que la probabilidad total de coincidencia es 0.38.



Por consiguiente, observar dos respuestas iguales no demuestra que se haya usado una regla determinista; para una distribución muy concentrada, las coincidencias pueden ser frecuentes, mientras que una distribución uniforme sobre muchas opciones puede producirlas con menor frecuencia, sin que esa diferencia mida por sí sola calidad o inteligencia.

La posibilidad inversa también importa: dos resultados diferentes no prueban que haya intervenido un sorteo; una función determinista puede devolver otra salida si se modifica el contexto o alguno de sus datos de entrada, de forma que el diagnóstico necesita evidencia sobre el mecanismo y las condiciones, no solamente sobre el texto final.

Los ejemplos permiten una conclusión acotada y útil: la variación observable es compatible con distintas causas y la igualdad observable también; el número de respuestas examinadas y el control de las condiciones determinan cuánto puede aprenderse de la comparación.



Elegir el siguiente paso no equivale a elegir toda la respuesta

Una respuesta de varias unidades se construye sobre una sucesión de contextos; después de elegir el primer elemento, ese elemento se incorpora a las condiciones del siguiente, por lo que comparar dos trayectorias exige seguir las probabilidades pertinentes en cada rama.

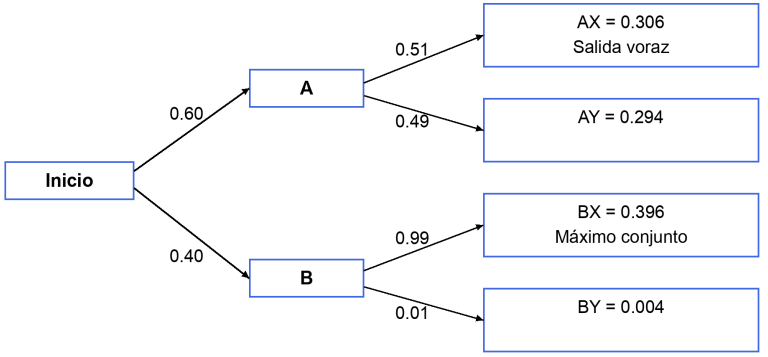
En un ejemplo hipotético de dos pasos, el primero permite A con probabilidad 0.60 o B con 0.40; si se eligió A, el segundo paso permite X con 0.51 o Y con 0.49, mientras que después de B permite X con 0.99 o Y con 0.01, y la secuencia termina siempre después de ese segundo paso.

La probabilidad de AX es 0.60 × 0.51 = 0.306 y la de AY es 0.60 × 0.49 = 0.294; la probabilidad de BX es 0.40 × 0.99 = 0.396 y la de BY es 0.40 × 0.01 = 0.004, de manera que las cuatro trayectorias suman uno y forman el universo completo de este pequeño ejemplo.

Si se elige vorazmente en cada paso, se toma A por superar a B y luego X por superar a Y, obteniendo AX; sin embargo, BX tiene mayor probabilidad conjunta, lo cual demuestra que maximizar cada decisión local no equivale

necesariamente a maximizar la probabilidad de la secuencia completa.

Figura 1. *Árbol hipotético que distingue máximo local y máximo conjunto*



Nota. Elaboración propia; cada valor terminal es el producto de las probabilidades de su trayectoria, AX es la salida voraz y BX es la secuencia más probable del universo definido.

El ejemplo no afirma que una búsqueda más extensa resuelva automáticamente cualquier tarea de escritura; sólo establece una diferencia matemática entre dos



objetivos, porque «mayor probabilidad de secuencia» todavía no incluye una definición de exactitud factual, utilidad, claridad o adecuación a una persona concreta.

Holtzman et al. (2020) compararon estrategias de decodificación en generación abierta y documentaron problemas de repetición y diversidad asociados a métodos de maximización en sus condiciones experimentales; ese resultado respalda la necesidad de distinguir un objetivo probabilístico de la calidad del texto, pero no autoriza a declarar que toda selección determinista sea inferior en cualquier tarea.

Un sistema que debe extraer un código de un catálogo puede valorar la estabilidad de manera diferente a un sistema que propone variantes de una explicación; en ambos casos conviene fijar el criterio de aceptación antes de juzgar el mecanismo, pues la diversidad tiene sentido respecto de un propósito y no como una virtud universal.

La longitud y la probabilidad conjunta

Al multiplicar probabilidades menores que uno, la probabilidad de una trayectoria específica puede hacerse



pequeña con rapidez; por ejemplo, una secuencia hipotética de diez pasos con probabilidad condicional 0.80 en cada paso tiene probabilidad conjunta 0.80^{10} = 0.1073741824, aun cuando cada elección individual sea relativamente favorecida.

Ese cálculo no significa que una respuesta de diez unidades tenga apenas 10.7 % de probabilidad de ser verdadera; calcula la probabilidad de una trayectoria definida bajo condiciones estipuladas, y nada en el ejemplo establece una relación entre las unidades generadas y la verdad de una proposición.

Tampoco permite comparar sin ajustes cualquier texto corto con cualquier texto largo, porque la longitud puede formar parte del evento y de las reglas de terminación; una evaluación seria debe precisar si compara secuencias de igual longitud, promedios por unidad o cumplimiento de una tarea, en lugar de usar indistintamente todos esos indicadores.

La enseñanza práctica es conservar el denominador conceptual: antes de interpretar un porcentaje, conviene preguntar qué resultados se contaron como posibles y qué



condiciones se fijaron; sin esa información, números técnicamente calculados pueden sostener conclusiones que nunca estuvieron incluidas en el cálculo.

Semillas y secuencias seudoaleatorias

La palabra «aleatorio» puede referirse a un modelo matemático de incertidumbre o al procedimiento computacional que genera números para simularlo; en muchos programas se emplean generadores seudoaleatorios, cuyos estados evolucionan mediante una regla y cuya salida puede repetirse al restaurar las mismas condiciones de inicio.

Una semilla es un dato utilizado para inicializar ese proceso, pero no es una descripción completa del programa; conservarla sólo puede ayudar a reproducir aquello que depende del generador correspondiente, siempre que se mantengan las demás condiciones necesarias y que el flujo de llamadas sea compatible.

La política de compatibilidad de NumPy especifica condiciones estrictas para repetir un flujo: mismo generador de bits, semilla, secuencia de llamadas,



argumentos, compilación, entorno y máquina; también advierte que cambiar el tamaño de una solicitud puede cambiar el algoritmo utilizado, de modo que «misma semilla» no constituye por sí sola una promesa de igualdad universal (NumPy Developers, s.f.).

Este ejemplo documental permite identificar un principio general de diagnóstico, sin atribuir NumPy a ninguna aplicación concreta: el significado de una semilla depende del mecanismo que la interpreta, por lo que el mismo número no tiene una semántica universal entre programas.

Un generador didáctico completamente visible

Para mostrar la relación entre regla determinista y apariencia variable, definamos un generador de juguete mediante la operación «multiplicar el estado por cinco, sumar uno y conservar el residuo al dividir entre dieciséis»; si el estado inicial es uno, el siguiente es seis porque 5 × 1 + 1 = 6, y el posterior es quince porque 5 × 6 + 1 = 31, cuyo residuo es quince.

Los ocho primeros estados generados son 6, 15, 12, 13, 2, 11, 8 y 9; al dividirlos entre dieciséis se obtienen 0.3750,



0.9375, 0.7500, 0.8125, 0.1250, 0.6875, 0.5000 y 0.5625, una secuencia irregular a simple vista que está completamente determinada por la regla y el estado inicial.

Si esos números se introducen en los intervalos del ejemplo A, B y C, producen A, C, B, C, A, B, B y B; repetir exactamente el estado inicial y las operaciones reproduce la secuencia, mientras que comenzar a consumir los números una posición después modifica los resultados observados aunque la semilla original no haya cambiado.

El generador se ofrece exclusivamente como demostración aritmética, no como recomendación para simular un modelo de lenguaje ni para proteger información; tiene un espacio de estados diminuto y una estructura visible, suficientes para comprender la idea, pero inadecuados para trasladar conclusiones sobre calidad estadística o seguridad a herramientas reales.

Además, su salida pertenece a una cuadrícula de dieciséis valores posibles, por lo que no implementa exactamente cualquier reparto decimal: recorrer un ciclo completo asignaría ocho valores a A, cinco a B y tres a C con los umbrales definidos, equivalentes a 0.50, 0.3125 y 0.1875;

esta diferencia evita presentar una ilustración finita como si fuera una muestra uniforme continua perfecta.

La tabla 2 permite revisar las primeras operaciones sin depender de la autoridad del texto; basta con aplicar la regla de residuos y comprobar después el intervalo al que pertenece cada cociente.

Tabla 2. *Primeros pasos del generador didáctico con estado inicial uno*

Paso	Estado generado	Estado dividido entre 16	Categoría
1	6	0.3750	A
2	15	0.9375	C
3	12	0.7500	B
4	13	0.8125	C
5	2	0.1250	A
6	11	0.6875	B
7	8	0.5000	B

Paso	Estado generado	Estado dividido entre 16	Categoría
8	9	0.5625	B

Nota. Elaboración y cálculo propios; A corresponde a [0, 0.50), B a [0.50, 0.80) y C a [0.80, 1), donde el corchete incluye el extremo y el paréntesis lo excluye.

Repetir una ejecución y explorar otras posibles

Restaurar un estado sirve para investigar un fallo porque permite comparar cambios del programa sobre una trayectoria controlada; sin embargo, evaluar un sistema sólo con esa trayectoria puede ocultar problemas que aparecen bajo otras elecciones, por lo que el control experimental y la exploración de variabilidad cumplen funciones complementarias.

En el ejemplo didáctico, una persona que publique únicamente la primera salida informará A y otra que observe las ocho primeras encontrará también B y C; ninguna descripción contradice la regla, pero cada una cubre una parte diferente del comportamiento, de manera



que el alcance de una prueba debe quedar unido a su protocolo.

La documentación de PyTorch distingue controlar fuentes de aleatoriedad y evitar algoritmos no deterministas, además advierte que la reproducibilidad completa no está garantizada entre versiones, plataformas o ejecuciones en CPU y GPU; sus recomendaciones se refieren a configuraciones concretas y reconocen posibles costos de rendimiento de las operaciones deterministas (PyTorch Contributors, 2026).

La inferencia razonable es limitada: una semilla puede formar parte de un expediente de repetición, pero necesita acompañarse de información sobre software y ejecución; si un servicio no expone esos datos, la falta de acceso restringe lo que puede afirmarse y obliga a describir la repetibilidad como observada bajo ciertas condiciones, en lugar de presentarla como propiedad absoluta.

La ejecución numérica también forma parte del resultado

El determinismo de una regla abstracta no resuelve por completo la repetibilidad de su implementación; una



operación escrita sobre números reales puede ejecutarse con representaciones finitas, redondeos y órdenes de cálculo específicos, de modo que el programa concreto requiere una descripción más detallada que la fórmula conceptual.

Goldberg (1991) explica por qué muchos números reales no pueden representarse exactamente con un número finito de bits y por qué los resultados deben redondearse; también muestra que la suma en coma flotante no conserva siempre la asociatividad, propiedad que en aritmética exacta permite cambiar la agrupación de sumandos sin cambiar el resultado.

Para ilustrarlo sin atribuir una prueba a una biblioteca determinada, imaginemos una aritmética decimal que conserva tres cifras significativas después de cada suma; la agrupación (1000 + 1) + (−1000) redondea primero 1001 a 1000 y termina en cero, mientras que 1000 + (1 + (−1000)) calcula primero −999 y termina en uno.

Ambas evaluaciones siguen reglas determinadas, pero no ejecutan la misma secuencia de redondeos; por tanto, obtener resultados distintos al cambiar el orden de las

operaciones no demuestra una fuente de azar, aunque sí demuestra que la forma de implementación importa para el resultado calculado.

El ejemplo requiere mantener su propia condición: se trata de una aritmética simplificada a tres cifras significativas, no de una medición de procesadores ni de una predicción sobre la frecuencia de diferencias en un servicio; su función consiste en hacer visible un mecanismo que la aritmética exacta puede ocultar.

Cuándo una diferencia pequeña cambia la selección

No toda diferencia numérica modifica la salida final; si la mayor puntuación conserva una ventaja amplia sobre la segunda, una perturbación pequeña puede alterar decimales sin cambiar el elemento elegido, mientras que dos puntuaciones casi empatadas pueden invertir su orden con una modificación mucho menor.

Sea m la diferencia entre la puntuación más alta y la segunda, y supongamos que el error absoluto de cada puntuación está acotado por ε; en el peor caso, la primera puede disminuir ε y la segunda aumentar ε, por lo que la



ventaja mínima restante es m − 2ε y la selección del máximo permanece estable si m es estrictamente mayor que 2ε.

Esta condición es suficiente dentro del modelo simplificado, no una garantía disponible para cualquier IA; exige conocer una cota del error sobre todas las puntuaciones competidoras y mantener una regla de selección definida, requisitos que rara vez pueden comprobarse desde una conversación ordinaria con una aplicación cerrada.

En un caso hipotético con puntuaciones 4.20 y 4.10, el margen es 0.10 y una perturbación de hasta 0.01 por puntuación deja una ventaja mínima de 0.08; si las puntuaciones son 4.200 y 4.199, el margen es 0.001 y esa misma cota ya no asegura el orden, aunque tampoco obliga a que cambie.

La distinción entre «puede cambiar» y «cambiará» es fundamental: perder una garantía matemática sólo significa que los supuestos ya no excluyen un resultado alternativo; afirmar que se producirá un fallo requeriría



evidencia adicional sobre la perturbación realmente ocurrida.

De una diferencia local a una trayectoria distinta

Cuando cambia un token seleccionado, puede cambiar el contexto de los pasos posteriores y, por esa vía, la respuesta completa; esta propagación es una posibilidad del procedimiento autorregresivo, no una prueba de que cada diferencia microscópica produzca una divergencia visible o relevante para la tarea.

El árbol de la figura 1 permite entenderlo: pasar de A a B altera las probabilidades del segundo paso porque la condición es diferente; una vez separadas las ramas, comparar sus salidas como si procedieran del mismo contexto ya no describe correctamente el experimento.

La documentación técnica de PyTorch consultada señala diferencias de reproducción numérica entre implementaciones de operaciones, relacionadas entre otros factores con el orden de acumulación; este respaldo se limita a la biblioteca documentada y no permite identificar, sin registros adicionales, la causa de una



discrepancia observada en una aplicación externa (PyTorch Contributors, 2026).

Conviene, por ello, distinguir tres situaciones: selección estocástica prevista por diseño, variación derivada de condiciones de ejecución y cambio de entrada o estado; todas pueden producir textos diferentes, pero requieren medidas de diagnóstico distintas y no deben recibir una explicación única por comodidad.

Verdad y probabilidad de emisión

La probabilidad de que un sistema emita una frase y la verdad de esa frase son propiedades diferentes; una afirmación falsa puede ser frecuente en un conjunto de textos y una afirmación verdadera puede aparecer poco, de manera que la frecuencia de emisión no constituye por sí sola un criterio de comprobación.

Supongamos un catálogo hipotético que establece que el código de un expediente es R27; un generador puede asignar 0.90 a producir R72 y 0.10 a producir R27, por lo que elegir siempre la continuación más probable repetiría sistemáticamente el código incorrecto, mientras que



muestrear permitiría en ocasiones el correcto sin corregir la distribución subyacente.

Si después se invierten esas probabilidades, cambia la conveniencia de las decisiones bajo el criterio «reproducir el código del catálogo»; el ejemplo demuestra que la evaluación depende de relacionar resultados con una fuente de referencia, no de clasificar el mecanismo como aleatorio o determinista y dar por resuelta la tarea.

La construcción usa un hecho estipulado para que no haya ambigüedad: R27 es correcto porque así se definió el catálogo del caso, no porque sea un código real; en problemas del mundo, establecer la referencia puede exigir consultar documentos, resolver diferencias de fecha o distinguir una interpretación de un dato comprobable.

El alcance de la evidencia sobre respuestas falsas

Lin et al. (2022) diseñaron TruthfulQA para evaluar respuestas ante preguntas que pueden inducir la reproducción de errores y creencias equivocadas; su análisis documenta falsedades imitativas en los modelos estudiados y respalda la separación entre reproducir



patrones de lenguaje y responder verazmente, sin establecer una tasa universal de error para cualquier IA o consulta.

El estudio es relevante aquí por su objeto y su diseño, no porque sus resultados históricos permitan clasificar productos actuales; una nueva versión, un contexto distinto, una fuente recuperada o un método de evaluación diferente cambian las condiciones y requieren una comprobación propia.

También sería excesivo concluir que el carácter probabilístico vuelve imposible toda respuesta correcta; el catálogo hipotético ya permite construir distribuciones que favorecen la respuesta correcta, y un procedimiento de validación puede rechazar una salida incompatible con los datos disponibles, aunque siga siendo necesario examinar la cobertura y los límites del validador.

La corrección puede adoptar distintas formas según la tarea: una suma admite un resultado verificable bajo reglas definidas, una síntesis requiere fidelidad a su fuente y una propuesta creativa puede evaluarse por restricciones y utilidad; llamar «verdadero» a cualquier resultado

satisfactorio diluiría diferencias que el análisis debe conservar.

Confianza verbal y calibración

Cuando un sistema escribe «estoy muy seguro», produce una expresión lingüística; para convertir esa expresión en una estimación evaluable se necesita definir a qué evento se refiere, cómo se obtuvo el valor y con qué conjunto de casos se contrastó, pues el tono de seguridad no aporta por sí solo esas condiciones.

Guo et al. (2017) estudian la calibración en clasificación supervisada mediante la relación entre confianza declarada por el modelo y frecuencia de aciertos; la distinción es útil para comprender que una probabilidad puede necesitar evaluación empírica, pero sus experimentos no acreditan automáticamente la calibración de respuestas abiertas de un modelo de lenguaje.

En un ejemplo hipotético de cien decisiones comparables anunciadas con confianza de 0.80, observar ochenta aciertos sería compatible con la idea de calibración en ese grupo; sin embargo, ese resultado agregado no garantiza



que una decisión particular sea correcta ni demuestra que el comportamiento se mantenga para otro tema, otra población o una forma diferente de preguntar.

El mismo razonamiento impide convertir la probabilidad de un token en una confianza general del documento: una respuesta puede contener varias afirmaciones y algunas pueden depender de fuentes distintas; antes de calcular una confianza global hace falta definir cómo se relacionan esas afirmaciones con el evento que se desea evaluar.

Coincidencia entre respuestas y acuerdo con la evidencia

Repetir una consulta puede servir para detectar inestabilidad, pero seleccionar la respuesta más frecuente sólo mejora algo si se especifican supuestos que vinculan esa frecuencia con la corrección; cuando todas las ejecuciones comparten una fuente equivocada o una interpretación defectuosa, el acuerdo puede concentrarse precisamente en el error.

En el catálogo del ejemplo, cien respuestas R72 no derrotan al registro R27 por votación; la repetición muestra

un comportamiento estable del generador, mientras que la contradicción con la fuente revela el problema de contenido, y ambas observaciones pueden ser ciertas al mismo tiempo.

La utilidad de repetir consiste entonces en producir evidencia sobre variación bajo condiciones controladas; para comprobar hechos se necesita además evidencia externa pertinente, de modo que la multiplicación de respuestas y la verificación documental no son procedimientos intercambiables.

Cómo investigar dos respuestas diferentes

Un diagnóstico útil comienza conservando las dos respuestas y evitando explicaciones prematuras; a partir del texto visible puede comprobarse qué cambió, pero atribuir la causa exige datos sobre entrada, selección y ejecución, por lo que conviene distinguir la observación de la hipótesis desde el primer registro.

La comparación inicial debe separar variación superficial, variación de contenido y variación de conducta; cambiar el orden de dos ejemplos puede conservar la explicación,

sustituir una cifra puede alterar una conclusión y ejecutar una acción diferente puede afectar el resultado de todo el flujo, aunque los mensajes finales se parezcan.

Antes de repetir una prueba, se propone fijar el criterio de equivalencia: para una suma puede ser igualdad numérica, para una extracción puede ser coincidencia exacta con un campo y para una explicación puede ser conservación de un conjunto de afirmaciones; este criterio debe definirse antes de observar qué salida resulta más conveniente.

Caso de una respuesta que cambia de redacción

Imaginemos una prueba hipotética con cuarenta ejecuciones de la instrucción «explica por qué cinco más siete equivale a doce», bajo un entorno que el evaluador intenta mantener estable; veinticuatro respuestas empiezan descomponiendo siete, diez utilizan una recta numérica y seis agrupan unidades, mientras que todas conservan el resultado doce y los pasos aritméticos correctos.

La diversidad de redacción observada en ese supuesto no constituye cuarenta soluciones matemáticas



incompatibles; corresponde a formas diferentes de exponer el mismo contenido, por lo que penalizar cada diferencia textual como un error confundiría estabilidad de cadena con fidelidad conceptual.

Las proporciones descriptivas serían 60 %, 25 % y 15 % para los tres tipos de explicación; esos porcentajes resumen la muestra inventada del caso y no son estimaciones medidas de ningún modelo, además una clasificación distinta de las explicaciones produciría otro resumen sin cambiar necesariamente la corrección de las respuestas.

Si el objetivo fuera generar una ficha escolar con texto idéntico en cada impresión, la variación sí podría incumplir el requisito de formato; el mismo conjunto de respuestas sería aceptable para explorar explicaciones y problemático para una salida contractual de cadena fija, lo cual muestra por qué la tarea debe preceder al juicio.

Caso de una contradicción que exige una fuente

Consideremos ahora un inventario hipotético cuya fila Q5 contiene dieciocho piezas disponibles y cuya fecha de



corte está fijada; una ejecución informa dieciocho y otra informa ochenta y una, manteniendo idéntico tono de seguridad, de modo que la diferencia ya no puede tratarse como una paráfrasis.

El primer paso propuesto es cotejar ambas salidas con la fila y la fecha efectivamente consultadas; si el sistema recibió archivos diferentes, el incidente pertenece inicialmente al control de las entradas, mientras que si recibió el mismo dato y transformó mal la cantidad, la investigación debe avanzar hacia extracción, generación o validación.

Repetir la pregunta diez veces puede mostrar cuál respuesta reaparece, pero no sustituye ese cotejo; una prueba adicional útil consistiría en exigir que el sistema reproduzca el identificador de fila y el valor de origen, y comprobar ambos mediante un procedimiento independiente, sin asumir que una cita generada acredita por sí misma una consulta real.

La medida correctiva depende del hallazgo: mantener una copia identificada de la fuente ayuda a controlar su versión, verificar el campo puede detectar una transposición y



restringir el formato puede facilitar la comprobación; disminuir la variación de redacción no corrige necesariamente ninguno de esos fallos.

Caso de igualdad que oculta una debilidad

En un tercer escenario hipotético, una aplicación responde siempre R72 al consultar el código cuyo registro dice R27; la consistencia exacta entre cien ejecuciones podría aprobar una prueba de regresión puramente textual, aunque el sistema falle todas las veces respecto del criterio documental de corrección.

Una prueba de regresión compara una versión con un comportamiento esperado y puede detectar cambios, pero su valor depende de que ese comportamiento esperado sea pertinente; congelar una salida errónea conserva un defecto con gran disciplina técnica, por lo que el conjunto de pruebas necesita controles de contenido además de controles de estabilidad.

La objeción «al menos siempre se equivoca igual» reconoce una ventaja limitada para investigar el incidente, porque facilita reproducirlo; esa ventaja instrumental no

reduce por sí sola el daño de utilizar el dato incorrecto, y sería impropio convertirla en aprobación de la respuesta.

Un registro mínimo para distinguir causas

La tabla 3 propone datos que un evaluador puede conservar cuando dispone de acceso suficiente; no todos estarán disponibles en una interfaz comercial, de modo que los campos desconocidos deben mantenerse como desconocidos y no completarse con conjeturas sobre la infraestructura del proveedor.

Tabla 3. *Información útil para investigar una discrepancia*

Nivel	Dato que conviene conservar	Pregunta de diagnóstico
Entrada	Mensajes completos, archivos y resultados de herramientas	¿Se repitió realmente la misma información?
Modelo	Identificador de versión y configuración expuesta	¿Se comparó la misma versión disponible?

Nivel	Dato que conviene conservar	Pregunta de diagnóstico
Selección	Regla de elección y estado aleatorio accesible	¿Había muestreo o una selección fija?
Ejecución	Biblioteca, dispositivo y precisión cuando sean conocidos	¿Cambió una condición numérica relevante?
Evaluación	Fuente de referencia y criterio de aceptación	¿La diferencia altera el cumplimiento de la tarea?

Nota. Elaboración propia; es una propuesta de registro, no una descripción de campos que cualquier servicio deba ofrecer ni un procedimiento que garantice localizar todas las causas.

El diagnóstico debería cambiar una condición por vez cuando sea posible, conservar los resultados desfavorables y registrar las limitaciones de acceso; si se



modifican simultáneamente el prompt, el modelo y la fuente, una mejora posterior puede ser real, pero la atribución de su causa seguirá abierta.

Cuando no se dispone de probabilidades internas, el análisis todavía puede distinguir resultados correctos e incorrectos y medir variación observable; lo que no puede hacer legítimamente es reconstruir valores exactos de la distribución o mecanismos de servidor a partir de unas pocas respuestas textuales.

Mitos y objeciones que requieren una respuesta precisa

El primer mito afirma que la IA elige palabras «al azar» como si todas valieran lo mismo; el ejemplo de los intervalos muestra que un muestreo puede favorecer unas opciones sobre otras, por lo que una explicación adecuada necesita especificar la distribución y evita usar el azar uniforme como caricatura de cualquier generación probabilística.

Un segundo mito sostiene que una salida determinista es necesariamente confiable; el catálogo con el código



equivocado proporciona un contraejemplo completo, porque una función estable puede reproducir una incorrección y porque la confianza práctica requiere comprobar el contenido además de describir el mecanismo.

El tercer mito atribuye toda discrepancia a una semilla diferente; dos funciones deterministas pueden producir salidas distintas al recibir contextos diferentes y dos ejecuciones pueden variar por condiciones numéricas, de modo que la semilla sólo es una hipótesis entre varias hasta que existan registros que permitan aislarla.

También debe rechazarse la tesis inversa de que cualquier repetición prueba ausencia de azar; con la distribución de tres categorías, la probabilidad de coincidencia entre dos extracciones independientes es 38 %, una cantidad obtenida sin eliminar el muestreo ni asumir uniformidad.

La objeción de que todo programa es determinista

Puede objetarse que un programa ejecutado en una máquina con estado completamente especificado sigue reglas y, por tanto, hablar de aleatoriedad sería sólo una



forma de expresar ignorancia; esta observación describe bien ciertos generadores seudoaleatorios, pero no elimina la utilidad de distinguir el modelo estadístico utilizado y el comportamiento observable del sistema completo.

Si un evaluador trata una semilla desconocida como variable y estudia resultados sobre muchas inicializaciones, el objeto de su análisis es una distribución de ejecuciones; si otro conserva el estado y reconstruye una trayectoria exacta, el objeto es una ejecución particular, y ambos pueden describir correctamente el mismo procedimiento desde niveles diferentes.

Para el propósito de este artículo no es necesario resolver una controversia metafísica sobre la naturaleza última del azar; basta con declarar si se habla de un sorteo ideal, de un generador seudoaleatorio o de variación observada cuya causa no se ha identificado, porque mezclar esos usos sí impide investigar un problema técnico concreto.

La objeción de que quitar el azar elimina el error

Eliminar el muestreo puede reducir una fuente de variación, pero conserva los errores que ya favorece la



distribución o que introducen las entradas y las reglas de procesamiento; el caso R27 frente a R72 demuestra esta limitación sin necesitar una hipótesis sobre intenciones o capacidades humanas del sistema.

La afirmación contraria, según la cual agregar azar corrige necesariamente un sesgo, tampoco se sostiene: explorar alternativas puede ayudar sólo si existe un procedimiento capaz de reconocer o aprovechar las mejores, mientras que seleccionar una alternativa diferente sin comprobarla puede sustituir un error por otro.

En términos prácticos, generar varias respuestas y evaluarlas es un procedimiento distinto de generar varias respuestas y elegir la más llamativa; la primera estrategia necesita criterios, fuentes y costos definidos, y la segunda puede aumentar la impresión de variedad sin mejorar la relación con los hechos.

Lo que aún no puede concluirse

La revisión no determina con qué frecuencia una aplicación cerrada cambia por causas de implementación, ni qué proporción de sus discrepancias procede de



muestreo; para responderlo harían falta registros del servicio, pruebas controladas y acceso a condiciones que este artículo no ha observado.

Tampoco demuestra que una configuración sea óptima para toda tarea; el resultado depende del objetivo, de las restricciones de salida y del costo de los errores, de manera que una recomendación responsable debe referirse a un uso evaluado y no a una preferencia universal por estabilidad o diversidad.

La incertidumbre aquí delimitada es comprobable en principio: puede reducirse documentando versiones, preservando entradas y comparando resultados contra criterios independientes; cuando esas observaciones no están disponibles, la conclusión debe conservar su alcance limitado sin rellenar los vacíos con explicaciones intuitivas.

Implicaciones para utilizar y evaluar una IA

Para una tarea con respuesta verificable, conviene priorizar la fuente de referencia y el control del resultado; pedir una redacción más estable puede ser útil después,



pero no debería desplazar la comprobación del dato central, especialmente cuando varias salidas repiten una misma cifra sin identificar de dónde procede.

Cuando la tarea admite múltiples soluciones, como proponer formas de explicar una idea, es razonable evaluar diversidad y cumplimiento por separado; una salida puede ser diferente y valiosa si conserva las restricciones, mientras que otra puede parecer novedosa porque omitió precisamente el requisito que debía respetar.

En un trabajo que necesita poder reconstruirse, se propone conservar la entrada completa, la salida y los identificadores de versión disponibles, además de las fuentes consultadas y el criterio de aceptación; estos registros permiten describir lo que realmente ocurrió, aunque no otorguen control sobre componentes internos que el servicio no expone.

Para quienes diseñan evaluaciones, la unidad de análisis debe declararse antes de calcular tasas: token, secuencia, afirmación, documento o acción; una buena tasa de coincidencia textual no acredita una buena tasa de aciertos



factuales, y un documento correcto puede admitir distintas cadenas igualmente válidas.

Una evaluación con repetición debería incluir tanto casos sencillos de referencia como casos que exijan distinguir fuentes o condiciones; conviene informar la cantidad de ejecuciones, el procedimiento de selección de ejemplos y los fallos encontrados, porque escoger únicamente la mejor respuesta transforma la prueba en una demostración parcial del potencial, no en una descripción completa del comportamiento observado.

La repetición controlada y la comprobación externa resuelven preguntas complementarias: la primera permite observar estabilidad o variación bajo un protocolo, mientras que la segunda permite juzgar el cumplimiento de una tarea respecto de criterios independientes; conservar esa división evita que un informe técnico confunda facilidad de reproducción con validez sustantiva.



Conclusión

El carácter probabilístico de un modelo no basta para explicar cómo se selecciona cada respuesta ni para anticipar si será distinta de la anterior; una distribución puede alimentar una regla determinista o un muestreo con pesos desiguales, y una ejecución puede depender de condiciones que no están contenidas en el último mensaje visible.

Los ejemplos desarrollados establecen límites concretos a varias inferencias habituales: la opción más probable no tiene que ser casi segura, dos muestras pueden coincidir, la selección voraz puede perder la secuencia de mayor probabilidad y una semilla compartida no describe por completo un entorno de ejecución; ninguna de esas observaciones transforma la probabilidad de emitir un texto en prueba de su verdad.

La consecuencia práctica consiste en formular preguntas separadas sobre distribución, elección, ejecución y contenido, y reunir evidencia apropiada para cada una; cuando una respuesta cambia, corresponde identificar qué cambió y bajo qué condiciones, mientras que cuando

permanece idéntica todavía corresponde comprobar que resuelve correctamente la tarea.

Referencias

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model [Un modelo de lenguaje probabilístico neuronal]. *Journal of Machine Learning Research, 3*, 1137–1155. https://jmlr.org/papers/v3/bengio03a.html

Goldberg, D. (1991). What every computer scientist should know about floating-point arithmetic [Lo que todo informático debería saber sobre la aritmética de coma flotante]. *ACM Computing Surveys, 23*(1), 5–48. https://doi.org/10.1145/103162.103163

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks [Sobre la calibración de redes neuronales modernas]. En D. Precup & Y. W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning* (Vol. 70, pp. 1321–1330). PMLR. https://proceedings.mlr.press/v70/guo17a.html

Holtzman, A., Buys, J., Du, L., Forbes, M., & Choi, Y. (2020). *The curious case of neural text degeneration* [El curioso caso de la degeneración del texto neuronal] (Versión 2). arXiv. https://doi.org/10.48550/arXiv.1904.09751

Lin, S., Hilton, J., & Evans, O. (2022). TruthfulQA: Measuring how models mimic human falsehoods [TruthfulQA: Medición de cómo los modelos imitan falsedades humanas]. En S. Muresan, P. Nakov, & A. Villavicencio (Eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 3214–3252). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.229

NumPy Developers. (s.f.). *Compatibility policy* [Política de compatibilidad]. NumPy v2.5 Manual. Recuperado el 3 de octubre de 2026, de https://numpy.org/doc/stable/reference/random/compatibility.html

PyTorch Contributors. (2026, 14 de mayo). *Reproducibility* [Reproducibilidad]. PyTorch 2.14 documentation. https://docs.pytorch.org/docs/2.14/notes/randomness.html

