

**DECODIFICACIÓN ESPECULATIVA EN MODELOS DE INTELIGENCIA
ARTIFICIAL GENERATIVA AUTORREGRESIVA**

Referencia cruzada con solicitudes relacionadas

5

Esta solicitud reivindica prioridad a la solicitud de patente de los EE. UU. con N.º de serie 18/538.912, titulada “Speculative Decoding in Autoregressive Generative Artificial Intelligence Models”, presentada el 13 de diciembre de 2023, que reivindica el beneficio y prioridad de la solicitud de patente provisional de los EE. UU. con N.º de serie 63/460.850, titulada “Speculative Decoding in Autoregressive Generative Artificial Intelligence Models”, presentada el 20 de abril de 2023, ambas cedidas al cesionario de la presente y que se incorporan en la presente como referencia en su totalidad.

INTRODUCCIÓN

Aspectos de la presente divulgación se relacionan con modelos de inteligencia artificial generativa, y más específicamente con la decodificación especulativa en modelos de inteligencia artificial generativa (también referidos en la presente como “modelos de inteligencia artificial generativa”).

Los modelos de inteligencia artificial generativa se pueden usar en diversos entornos para generar una respuesta a un indicador de entrada (también denominado una consulta o una entrada). Por ejemplo, los modelos de inteligencia artificial generativa se pueden usar en aplicaciones de chatbot en las que se usan modelos de lenguaje grande (LLM, en inglés) para generar una respuesta, o al menos una respuesta, a un indicador de entrada. Otros ejemplos en los que se pueden usar modelos de inteligencia artificial generativa incluyen un modelo de difusión latente, en el que un modelo genera una imagen a partir de un texto de entrada descriptivo del contenido de la imagen deseada, transformadores de decisión, en el que se predicen acciones futuras basado en secuencias de acciones previas dentro de un determinado entorno, o lo similar.

Generalmente, generar una respuesta a una consulta usando modelos de inteligencia artificial generativa puede ser computacionalmente costoso. Por ejemplo, en una implementación de chatbot en la que se usa un modelo de lenguaje grande para generar una respuesta a una consulta formateada como una consulta de texto, se puede generar una respuesta a la consulta usando un pase a través del modelo de lenguaje grande para cada token (p. ej., una palabra o parte de una palabra) generado como parte de la respuesta. La salida de cada pase puede ser una distribución de probabilidad en un conjunto de tokens

(p. ej., palabras o partes de palabras) a partir de las cuales se puede seleccionar el siguiente token (p. ej., una palabra o parte de una palabra), por ejemplo, mediante muestreo o basado en la máxima probabilidad. Debido a que se usa un pase a través de un modelo de lenguaje grande para generar cada palabra (o token(s)) en una respuesta a una consulta, el gasto computacional puede modelarse como el producto del número de palabras incluidas en la respuesta y el gasto de recursos computacionales (p. ej., en términos de potencia de procesamiento, ancho de banda de memoria y/u otros recursos de cómputo usados) de realizar un paso a través del modelo de lenguaje grande, que generalmente aumenta a medida que el número de parámetros dentro del modelo de lenguaje grande aumenta.

15

SUMARIO

Ciertos aspectos de la presente divulgación proporcionan un método para generar una respuesta a un indicador de entrada usando un modelo de inteligencia artificial generativa. El método de ejemplo incluye generalmente recibir una pluralidad de conjuntos de tokens generados basado en un indicador de entrada y un primer modelo de inteligencia artificial generativa, cada conjunto de tokens en la pluralidad de conjuntos de tokens correspondiente a una respuesta candidata al indicador de

entrada; seleccionar, usando un segundo modelo de inteligencia artificial generativa y ajuste recursivo de una distribución objetivo asociada con la pluralidad recibida de conjuntos de tokens, un conjunto de tokens de la pluralidad de conjuntos de tokens; y emitir el conjunto de tokens seleccionado como una respuesta al indicador de entrada.

Ciertos aspectos de la presente divulgación proporcionan un método para generar una respuesta a un indicador de entrada usando un modelo de inteligencia artificial generativa. El método generalmente incluye generar, basado en un indicador de entrada y un modelo de inteligencia artificial generativa, una primera pluralidad de conjuntos de tokens, cada conjunto de tokens en la primera pluralidad de conjuntos de tokens correspondiente a una primera porción de una respuesta candidata al indicador de entrada. Usando el modelo de inteligencia artificial generativa, se genera especulativamente una segunda pluralidad de conjuntos de tokens. Cada conjunto de tokens en la segunda pluralidad de conjuntos de tokens corresponde generalmente a una segunda porción de la respuesta candidata al indicador de entrada basada en la primera pluralidad de conjuntos de tokens. Mientras que se genera especulativamente la segunda pluralidad de conjuntos de tokens, se selecciona un conjunto de tokens de la primera pluralidad de conjuntos de tokens, y el conjunto de tokens seleccionado de la primera pluralidad de tokens y un

conjunto de tokens asociado en la segunda pluralidad de tokens se emiten como una respuesta al indicador de entrada.

Otros aspectos proporcionan sistemas de procesamiento configurados para realizar los métodos mencionados anteriormente así como los divulgados en la presente; medio legible por computador no transitorio que comprende instrucciones que, cuando son ejecutadas por uno o más procesadores de un sistema de procesamiento, hacen que el sistema de procesamiento realice los métodos mencionados anteriormente así como los divulgados en la presente; un producto de programa informático incorporado en un medio de almacenamiento legible por computador que comprende un código para realizar los métodos mencionados anteriormente, así como los divulgados además en la presente; y un sistema de procesamiento que comprende medios para ejecutar los métodos mencionados anteriormente así como los divulgados además en la presente.

La siguiente descripción y los dibujos relacionados establecen en detalle ciertas características ilustrativas de uno o más aspectos.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

Las Figuras anexas representan únicamente ciertos aspectos de esta divulgación y, por lo tanto, no deben considerarse limitantes del alcance de esta divulgación.

La Figura 1 ilustra un ejemplo de decodificación especulativa en modelos de inteligencia artificial generativa, de acuerdo con aspectos de la presente divulgación.

5 **Las Figuras 2A y 2B** ilustran ejemplos de decodificación especulativa recursiva en modelos de inteligencia artificial generativa, de acuerdo con aspectos de la presente divulgación.

La Figura 3 ilustra un árbol de ejemplo de tokens generados usando decodificación especulativa recursiva en modelos de inteligencia artificial generativa, de acuerdo con aspectos de la presente divulgación.

La Figura 4 ilustra operaciones de ejemplo para generar una respuesta a un indicador de entrada usando decodificación especulativa recursiva en modelos de inteligencia artificial generativa, de acuerdo con aspectos de la presente divulgación.

La Figura 5 ilustra un ejemplo de canalización para decodificación autoespeculativa en modelos de inteligencia artificial generativa, de acuerdo con aspectos de la presente divulgación.

La Figura 6 ilustra arquitecturas de ejemplo para decodificación autoespeculativa en modelos de inteligencia artificial generativa, de acuerdo con aspectos de la presente divulgación.

La Figura 7 ilustra operaciones de ejemplo para generar una respuesta a un indicador de entrada usando un modelo de inteligencia artificial generativa y decodificación autoespeculativa, de acuerdo con aspectos
5 de la presente divulgación.

La Figura 8 representa un sistema de procesamiento de ejemplo configurado para realizar diversos aspectos de la presente divulgación.

La Figura 9 representa un sistema de
10 procesamiento de ejemplo configurado para realizar diversos aspectos de la presente divulgación.

Para facilitar la comprensión, se han usado números de referencia idénticos, en la medida de lo posible, para designar elementos idénticos que son comunes
15 a las figuras. Se contempla qué elementos y características de un aspecto puedan incorporarse beneficiosamente en otros aspectos sin mayor mención.

DESCRIPCIÓN DETALLADA

Aspectos de la presente divulgación proporcionan aparatos, métodos, sistemas de procesamiento y medios legibles por computador para generar eficientemente respuestas a consultas de entrada usando modelos de inteligencia artificial generativa.

Generalmente, los modelos de inteligencia artificial generativa generan una respuesta a una consulta de entrada en el modelo. Por ejemplo, un modelo de lenguaje grande (LLM) desplegado dentro de un chatbot puede generar una respuesta a una consulta usando múltiples pases a través del modelo de lenguaje grande, siendo cada pase sucesivo basado en la consulta (que puede tokenizarse para su procesamiento) y los tokens (o palabras) generados usando pasadas anteriores a través del modelo de lenguaje grande. En general, estos modelos lingüísticos grandes pueden incluir un gran número (p. ej., miles de millones o incluso billones) de pesos o parámetros dentro del modelo. Debido al tamaño de estos modelos y las operaciones realizadas en cada token para predecir cuál debería ser el siguiente token generado en respuesta a una consulta y los tokens generados previamente, puede no ser práctico, o incluso posible, implementar modelos de lenguaje grandes en una variedad de dispositivos que tienen capacidades limitadas de memoria, almacenamiento y/o procesamiento en relación con las instancias de computación

en la nube en las que normalmente operan modelos de lenguaje grande. Además, en algunos casos, el ancho de banda de memoria involucrado en la generación de una respuesta a una consulta proporcionada como entrada en un modelo puede evitar que los
5 recursos informáticos se utilicen para otras tareas.

Para mejorar la eficiencia y el rendimiento de los modelos de lenguaje grande, las técnicas de decodificación especulativa permiten que un modelo de lenguaje más pequeño, a veces conocido como modelo de lenguaje grande de borrador
10 (o como un modelo de borrador o un modelo de aproximación), se ejecute (p. ej., secuencialmente o en paralelo) con un modelo de lenguaje más grande, a veces conocido como modelo de lenguaje grande objetivo (o como un modelo objetivo). En tal caso, el modelo de borrador puede generar tokens
15 especulativamente adicionales en secuencia y probabilidades usadas para muestrear estos tokens adicionales basado en un conjunto actual de tokens aceptados. El modelo objetivo puede generar tokens basado en los tokens generados por el modelo de borrador. Para generar un resultado, el modelo objetivo
20 puede realizar un muestreo de rechazo por token para aceptar o rechazar tokens individuales generados por el modelo de borrador de tal manera que el modelo de borrador y el modelo objetivo tengan distribuciones de probabilidad lo similar.

En algunos aspectos, el modelo de borrador puede
25 ser una versión podada del modelo objetivo elegido de tal manera que el modelo de borrador y el modelo objetivo

tengan distribuciones de probabilidad lo similar. En otros aspectos, el modelo de borrador puede ser una versión más pequeña del modelo objetivo (p. ej., entrenado en millones de tokens, en lugar de cientos de millones o incluso miles de millones de tokens).

Ciertos aspectos de la presente divulgación proporcionan técnicas y aparatos para generar respuestas a una entrada de consulta en un modelo de lenguaje grande usando técnicas de decodificación especulativa recursiva.

10 Generalmente, el modelo de borrador puede generar uno o más conjuntos de tokens como respuestas candidatas a la consulta, que pueden estructurarse como una pluralidad de ramas (p. ej., en una estructura de datos de árbol). El modelo objetivo, a su vez, puede realizar el rechazo de

15 muestreo recursivamente en tokens proporcionados por el modelo de borrador. El rechazo de muestreo generalmente recursivo permite que la distribución de probabilidad usada por el modelo objetivo en los tokens de muestreo generados por el modelo de borrador se actualice para eliminar un

20 token rechazado por el modelo objetivo, y la distribución de probabilidad actualizada se usa después para muestrear tokens posteriores generados por el modelo de borrador. Al realizar el muestreo de rechazo de manera recursiva, ciertos aspectos de la presente divulgación pueden

25 conservar una estrecha relación entre las distribuciones de probabilidad dentro del borrador y los modelos objetivo,

al tiempo que aumentan el rendimiento de los modelos de borrador y objetivo (p. ej., el número de tokens generados por segundo) en relación con los modelos de borrador y objetivo configurados para generar respuestas por token.

5 Otros aspectos de la presente divulgación proporcionan técnicas y aparatos para generar respuestas a una entrada de consulta en un modelo de lenguaje grande usando técnicas de decodificación especulativa en las que un único modelo genera especulativamente tokens en respuesta a la
10 entrada de consulta y verifica tokens generados previamente, también referidos en la presente como “decodificación autoespeculativa”. En las técnicas de decodificación autoespeculativa, un modelo puede generar especulativamente uno o más tokens y generar especulativamente tokens
15 adicionales basado en números variables de tokens generados especulativamente que son verificados por el modelo. Al usar el mismo modelo para generar especulativamente tokens en respuesta a una consulta y para realizar la verificación de (p. ej., muestreo de rechazo en) el token generado
20 especulativamente, aspectos de la presente divulgación pueden reducir el gasto computacional involucrado en el entrenamiento y el uso de modelos de inteligencia artificial generativa en relación con el uso de múltiples modelos entrenados por separado para generar especulativamente tokens
25 y realizar la verificación de los tokens generados especulativamente. Además, la velocidad a la que se generan

los tokens puede maximizarse, o al menos incrementarse, con decodificación autoespeculativa en comparación con otras técnicas de decodificación especulativa.

5 Decodificación especulativa en modelos de inteligencia artificial generativa

Generalmente, la generación de tokens autorregresivos (p. ej., en modelos de lenguaje grande) puede tomar tokens históricos como entrada para generar una salida. Es decir, la generación de tokens autorregresivos puede estar representada por la expresión:

$$x_t \sim p(x|x_0, x_1, \dots, x_{t-1}) \rightarrow x_{t+1} \sim p(x|x_0, x_1, \dots, x_{t-1}, x_t)$$

donde x_t representa una secuencia de tokens generados en el tiempo t , teniendo una probabilidad condicional p condicionada a la selección de tokens x_0 a través x_{t-1} , y x_{t+1} representa una secuencia de tokens generados en el tiempo $t + 1$, teniendo una probabilidad condicional p condicionada a la selección de tokens x_0 a través de x_t . Generalmente, se puede generar un único token cada vez que se ejecuta un modelo autorregresivo, lo que significa que se pueden realizar N inferencias para generar una secuencia de N tokens. Como se analizó anteriormente, las técnicas de decodificación especulativa pueden usarse para acelerar la generación de

tokens mediante el uso de un modelo de borrador, más pequeño en tamaño que el modelo objetivo, que genere especulativamente tokens más rápido que el modelo objetivo, siendo el modelo objetivo usado para verificar los tokens (especulativamente) generados por el modelo de borrador.

En una canalización de decodificación especulativa, el modelo de borrador puede generar especulativamente n tokens autorregresivamente, de acuerdo con la expresión:

10

$$x_{t+1}^{draft} \sim p_t^{draft} = p^{draft}(x|x_0, \dots x_t), x_{t+2}^{draft} \sim p_{t+1}^{draft}, \dots, x_{t+n}^{draft} \sim p_{t+n-1}^{draft}$$

donde t corresponde a un punto en el tiempo, p_t^{draft} corresponde a la distribución de probabilidad condicional asociada a un token seleccionado en el x momento t condicionado a la selección de tokens x_0 a través x_{t-1} , y x_t^{draft} representa un token generado x especulativamente en el momento t por el modelo de borrador.

El modelo objetivo toma los n tokens generados y procesa los n tokens en paralelo para generar distribuciones de probabilidad para cada uno de los n tokens, de acuerdo con la expresión:

25

$$[p_t^{target}, p_{t+1}^{target}, \dots, p_{t+n}^{target}] = [p^{target}(x|x_0, \dots x_t, x_{t+1}^{draft}, \dots, x_{t+k}^{draft})]_{k=1,2,\dots,n}$$

donde k corresponde a un índice de token relativo a los n tokens generados y p_t^{target} corresponde a una distribución de probabilidad generada por el modelo objetivo en el momento t para los tokens x generados por el modelo de borrador.

5 Después, el modelo objetivo puede verificar los tokens generados por el modelo de borrador comparando distribuciones del modelo de borrador y el modelo objetivo para determinar si un token es aceptado o rechazado. Un token determinado x_{t+k}^{draft} puede aceptarse cuando
 10 $f(p_k^{draft}, p_k^{target}) < \alpha$, para alguna función f y algún umbral α (también conocido como tasa de aceptación). De cualquier otra manera, el token puede rechazarse. El token final puede generarse entonces en la primera posición de rechazo o en la última posición n basado en alguna función $g(p_k^{draft}, p_k^{target})$.

15 La decodificación especulativa, con una tasa de aceptación de α , puede resultar en reducciones de costos en relación con el uso de un único modelo autorregresivo para generar tokens iterativamente. Los ahorros de costos de inferencia, en relación con la generación iterativa de
 20 tokens, pueden representarse mediante la expresión:

$$C^{AR} = NC^{target} \rightarrow C^{SD} = nNC^{draft} + \frac{N}{\alpha + 1} C^{target}$$

donde N corresponde a un número de tokens, C^{AR} corresponde a un costo computacional usando una tasa de aceptación de α , C^{target} corresponde a un costo computacional de generar un conjunto de tokens usando el modelo objetivo, C^{draft} corresponde a un costo computacional de generar un conjunto de tokens usando el modelo de borrador, C^{SD} corresponde a un costo computacional de generar especulativamente un conjunto de tokens usando el modelo de borrador, y n corresponde a un número de tokens generados especulativamente en un único pase a través de un modelo autorregresivo. Consideremos un ejemplo en el que $N = 1000$, $C^{target} = 10$, $C^{draft} = 1$, $n = 4$, and $\alpha = 3$. En tal ejemplo, la decodificación especulativa puede resultar en una reducción del 35 % en el gasto computacional en relación con la generación iterativa autorregresiva de tokens por sí sola.

Sin embargo, la decodificación especulativa por token, como se analizó, puede imponer límites a la velocidad a la que se generan los tokens, ya que un primer token puede muestrearse individualmente por un modelo de borrador y después verificado por un modelo objetivo antes de que el siguiente token sea muestreado por el modelo de borrador y verificado por el modelo objetivo. Es decir, generar una respuesta a un indicador de entrada usando técnicas de decodificación especulativa por token puede implicar ejecutar el modelo de borrador y el modelo objetivo para cada token

generado como parte de una respuesta al indicador de entrada, que puede usar cantidades significativas de recursos computacionales (p. ej., tiempo de procesador, memoria, ancho de banda de memoria, etc.) para generar la respuesta.

5

Ejemplo de decodificación especulativa recursiva en modelos de inteligencia artificial generativa

La Figura 1 ilustra un ejemplo 100 de decodificación especulativa recursiva en modelos de inteligencia artificial generativa, de acuerdo con aspectos de la presente divulgación.

Como se ilustra, un modelo de borrador y un modelo objetivo pueden usarse en conjunto (o de cualquier otra manera juntos) para realizar decodificación especulativa recursiva de tokens para generar una respuesta a una consulta recibida para su procesamiento por uno o más modelos de inteligencia artificial generativa. Como se analiza con más detalle a continuación, la decodificación especulativa recursiva de tokens puede permitir que múltiples conjuntos de tokens (o secuencias de tokens) sean generados especulativamente por un modelo de borrador para su verificación por un modelo objetivo. Debido a que se pueden generar múltiples conjuntos (o secuencias) de tokens mediante un modelo de borrador para su verificación por un modelo objetivo, la decodificación especulativa recursiva puede aumentar la tasa de generación de tokens de modelos

de inteligencia artificial generativa al generar múltiples conjuntos (secuencias) de tokens que pueden aceptarse como una respuesta correcta, ya que un mayor número de conjuntos de tokens puede aumentar la probabilidad de que al menos
5 un conjunto incluya una secuencia de uno o más tokens que se aceptarán como respuesta.

El modelo de borrador generalmente selecciona una pluralidad de nodos de alta probabilidad (tokens) de una distribución de probabilidad para una salida sobre un
10 conjunto de tokens potenciales, dada una entrada de la consulta recibida. Los nodos de alta probabilidad (tokens) pueden seleccionarse basado en diversas técnicas, tales como selección superior- k (p. ej., selección de los k tokens que tienen las probabilidades más altas dentro de
15 una distribución de probabilidad), selección basada en núcleos (p. ej., selección basada en una suma de probabilidades que cumplen una probabilidad umbral), o lo similar. Al elegir muchos tokens candidatos, el modelo de borrador puede muestrear tokens basado en la distribución
20 de probabilidad y organizar una estructura de árbol que puede atravesarse recursivamente, como se analiza con más detalle a continuación, para identificar un conjunto de tokens que son una salida adecuada para la entrada dada.

Cuando un grupo de tokens muestreado se introduce
25 en el modelo de borrador en la siguiente iteración de ejecución del modelo de borrador, los tokens en el grupo

de tokens muestreado se introducen en la ubicación de la muestra y se tratan de forma independiente. El resultado puede ser una estructura de datos de árbol 110, con un indicador como un nodo raíz 111 de la estructura de datos de árbol, y niveles subsiguientes dentro de la estructura de datos de árbol 110 que representan diferentes tokens (o grupos de tokens), combinados con cada una de las combinaciones de tokens previamente seleccionadas. En algún punto en el tiempo (p. ej., después de generar un árbol con una profundidad definida, correspondiente a una longitud máxima de una secuencia generada por el modelo de borrador), el modelo de borrador puede emitir la estructura de datos de árbol generada 110 al modelo objetivo para su procesamiento adicional. La estructura de datos de árbol 110 puede, en algunos aspectos, emitirse al modelo objetivo con agrupaciones y probabilidades de selección generadas por el modelo de borrador.

En algunos aspectos, el modelo de borrador puede configurarse para activar la generación de tokens por el modelo objetivo (y posterior decodificación especulativa) basado en diversos criterios. Estos criterios pueden incluir un criterio de complejidad o criterio de rendimiento, tal como un criterio de complejidad o rendimiento asociado con el tamaño de la estructura de datos de árbol generada 110. En algunos aspectos, estos criterios pueden incluir un criterio de tiempo asociado

con una cantidad de tiempo esperada para que el modelo objetivo genere un conjunto de tokens contra los cuales se puede comparar la estructura de datos de árbol generada 110. En general, estos criterios de complejidad y/o
5 rendimiento pueden establecer un límite superior en el número de tokens generados por el modelo de borrador para su verificación por el modelo objetivo. Este límite superior puede, en algunos aspectos, estar basado en un número de nodos en la estructura de datos de árbol y puede
10 estar influenciado, por ejemplo, por un factor de ramificación definido para diferentes niveles de la estructura de datos de árbol 110 en la que se organizan los tokens muestreados, una profundidad de la estructura de datos de árbol 110, o lo similar. La carga computacional
15 en el peor de los casos en la última ronda de generación de tokens especulativos puede configurarse para estar vinculada por el ancho de banda de la memoria en el dispositivo en el que se ejecuta el modelo de borrador.

En algunos aspectos, el número de nodos en cada
20 nivel de la estructura de datos de árbol 110 (p. ej., donde el token n ilustrado en la **Figura 1** corresponde al nivel n en la estructura de datos de árbol 110, y donde el nivel 0 corresponde al nodo raíz 111 en la estructura de datos de árbol 110) y la profundidad de la estructura de datos de
25 árbol 110 puede definirse *a priori*. El número de nodos en cada nivel del árbol puede definirse globalmente, por nivel

o de alguna otra manera. Por ejemplo, el número de nodos en cualquier nivel de la estructura de datos de árbol 110 puede definirse basado en un factor de ramificación en el nivel anterior inmediato de la estructura de datos de árbol 110. Por ejemplo, un factor de ramificación de 2 para nodos en el nivel n de la estructura de datos de árbol 110 puede dar como resultado la generación de 2 nodos (tokens) en el nivel $n+1$ de la estructura de datos de árbol 110 para cada nodo (token) en el nivel n del árbol. Mientras tanto, la profundidad de la estructura de datos de árbol 110 puede definirse basada en un número máximo de tokens (p. ej., palabras) que se pueden generar usando cualquier pase a través del modelo de borrador. Por ejemplo, si el modelo de borrador se configura para generar una secuencia con una longitud máxima de 5 tokens durante cualquier instancia de generación especulativa, entonces la profundidad del árbol puede ser 6 (para incluir, en el primer nivel de la estructura de datos de árbol 110, el nodo raíz 111 correspondiente a la entrada en el modelo de borrador).

El modelo objetivo realiza recursivamente el muestreo de rechazo en (1) los tokens generados por el modelo de borrador e incluidos en la estructura de datos de árbol generada 110 y (2) una distribución de probabilidad q proporcionada como entrada al modelo objetivo. El muestreo de rechazo puede realizarse recursivamente en cada nodo del árbol generado, donde una condición de terminación de

selección de token por el modelo objetivo en una capa dada de la estructura de datos de árbol 110 se modela como un problema de recursión en el que una condición de terminación del problema de recursión es la aceptación de un token o el rechazo de todos los tokens. Al realizar el muestreo de rechazo recursivamente, el modelo objetivo puede aceptar o rechazar un token y ajustar la distribución de probabilidad usada para verificar un token posterior en el árbol generado. Si se rechaza un token, se puede generar una distribución de probabilidad actualizada $q' = (q - p)$ para el uso en la evaluación de tokens posteriores en el árbol, donde p representa la probabilidad asociada con el token rechazado de la distribución de probabilidad original q . Posteriormente, la distribución de probabilidad actualizada q' se puede usar para evaluar el siguiente token en el árbol. El conjunto de tokens seleccionado resultante 112 puede identificarse recursivamente, atravesando desde el nodo raíz 111 de la estructura de datos de árbol generada 110 basada en las distribuciones de probabilidad actualizadas generadas para cada nodo en la estructura de datos de árbol generada, como se analiza con más detalle a continuación.

En un ejemplo, el muestreo de rechazo recursivo se puede realizar usando técnicas “codiciosas”, en las que el primer token que se acepta se devuelve como una porción válida de una respuesta a la consulta de entrada (o

indicador). En otro ejemplo, se puede realizar un muestreo de rechazo recursivo para determinar si se acepta cada token en un nivel dado en la estructura de datos de árbol 110. El token seleccionado en ese nivel dado en la estructura de datos de árbol 110 puede ser, por ejemplo, el token que tiene la mayor probabilidad de ser un token válido para su inclusión en una respuesta al indicador de entrada. En otro ejemplo, se puede generar una distribución de probabilidad acumulativa para cada secuencia aceptada de tokens de la estructura de datos de árbol generada, y la secuencia con la distribución de probabilidad acumulada más grande puede seleccionarse como la respuesta al indicador de entrada. Debe reconocerse, por supuesto, que estos no son sino ejemplos de técnicas basadas en las cuales se pueden seleccionar secuencias de tokens a partir de la estructura de datos de árbol 110, y se pueden contemplar otras técnicas para seleccionar secuencias de tokens basado en el muestreo de rechazo recursivo (y el correspondiente ajuste de la distribución de probabilidad q cuando se rechazan tokens).

En algunos aspectos, el modelo de borrador puede coincidir con la distribución de probabilidad del modelo objetivo, pero puede tener un rendimiento de inferencia más rápido que el modelo objetivo en el mismo hardware. Generalmente, los modelos más pequeños pueden generar muchos tokens especulativos, pero pueden tener una mayor probabilidad de que los tokens generados sean rechazados

por el modelo objetivo. Las técnicas de decodificación especulativa analizadas en la presente pueden abordar esta mayor probabilidad de rechazo de token, a expensas de aumentar el gasto computacional para secuencias más
5 largas. Finalmente, en el modelo de borrador, un parámetro de temperatura (generalmente usado en la presente para referirse a un parámetro que influye en la probabilidad de que el modelo de borrador seleccione un token (palabra) con una probabilidad menor) puede ajustarse para mejorar
10 el rendimiento de la decodificación especulativa recursiva. En algunos aspectos, el modelo de borrador puede afinarse para igualar, o al menos aproximarse, la distribución de probabilidad del modelo objetivo y maximizar, o al menos aumentar, la probabilidad de que los
15 tokens generados especulativamente (o secuencias de tokens) generados por el modelo de borrador sean aceptados como tokens válidos por el modelo objetivo.

En general, el rendimiento de generación de tokens para la decodificación especulativa recursiva puede
20 incrementarse en relación con la decodificación especulativa por token. Es decir, para cualquier divergencia Kullback-Leibler (KL) dada entre el modelo de borrador y el modelo objetivo, el número de tokens generados para cada ejecución del modelo objetivo puede
25 ser mayor para la decodificación especulativa recursiva que para la decodificación especulativa por token. La

divergencia KL generalmente mide cómo la distribución de probabilidad del modelo de borrador difiere de la distribución de probabilidad del modelo objetivo (tratando el modelo objetivo como la distribución de referencia).

5 Diferentes estrategias de selección (p. ej., tamaño de grupo, tokens adicionales, etc.) pueden tener diferentes características de complejidad computacional. En consecuencia, la selección de una estrategia para seleccionar tokens para la aceptación del rechazo usando
10 decodificación especulativa recursiva puede basarse en una compensación entre la complejidad computacional y el rendimiento, dados los parámetros delimitadores de ancho de banda de lectura para los modelos de borrador y objetivo y el rendimiento del hardware.

15 **Las Figuras 2A y 2B** ilustran los ejemplos 200A, 200B de decodificación especulativa recursiva en modelos de inteligencia artificial generativa, de acuerdo con aspectos de la presente divulgación.

El ejemplo 200A ilustrado en **la Figura 2A**
20 representa un ejemplo en el que uno de múltiples tokens generados por un modelo de borrador se acepta para su inclusión en una respuesta a un indicador 210. Como se ilustra, el modelo de borrador genera cuatro tokens propuestos X_1 220A, X_2 220B, X_3 220C y X_4 220D, con
25 probabilidades asociadas a p_1, p_2, p_3, p_4 partir de una distribución objetivo original. q Un modelo objetivo puede

examinar secuencialmente estos tokens para determinar si acepta o rechaza cada token. Como se ilustra, el modelo objetivo examina primero el token X_1 220A para determinar si el token 220A debe aceptarse o rechazarse como un token a ser devuelto como parte de una respuesta a una entrada dada (p. ej., el indicador 210). En este ejemplo, la distribución objetivo q_1 222A se puede establecer en la distribución objetivo original q , y el modelo objetivo puede determinar si acepta o rechaza el token X_1 220A basado en los criterios de selección 224A:

$$U_1 < \min\left\{1, \frac{q_1(X_1)}{p_1(X_1)}\right\}$$

donde U_1 representa un número aleatorio generado entre $[0,1]$.

Si el token X_1 220A es aceptado (no mostrado), entonces el token X_1 220A puede emitirse, y el análisis del modelo objetivo de los tokens propuestos generados por el modelo de borrador puede proceder a la exploración del árbol para analizar nodos (tokens) conectados al nodo representado por el token X_1 220A en el árbol generado.

De cualquier otra manera, si, como se ilustra, el token X_1 220A es rechazado, entonces el modelo objetivo puede proceder para determinar si el token X_2 220B debe ser aceptado o rechazado. Al hacerlo, el modelo objetivo

puede usar una nueva distribución objetivo q_2 222B, que puede ser el resultado de restar la probabilidad p_1 asociada con el token rechazado X_1 220A de la distribución objetivo q_1 222A, de tal manera que. $q_2 = (q_1 - p_1)^+$ El modelo
5 objetivo puede entonces determinar si acepta o rechaza el token X_2 220B basado en criterios de selección 224B:

$$U_2 < \min\left\{1, \frac{q_2(X_2)}{p_2(X_2)}\right\}$$

10 donde U_2 también representa un número aleatorio generado entre $[0,1]$.

Al igual que con el token X_1 220A analizado anteriormente, el modelo objetivo puede determinar que el token X_2 220B satisface los criterios de aceptación y así
15 devolver el token X_2 220B como el token seleccionado (no mostrado). De cualquier otra manera, si, como se ilustra, el token X_2 220B es rechazado, entonces el modelo objetivo procede a determinar si el token X_3 220C debe aceptarse o rechazarse, usando una distribución objetivo q_3 222C
20 actualizada que elimina p_2 de q_2 (p. ej., de tal manera que $q_3 = (q_2 - p_2)^+$) y los criterios de aceptación U_3 224C. Este proceso puede continuar hasta que, como se ilustra, se determine que un token (en este ejemplo, el token X_4 220D ilustrado en el ejemplo 200A usando una distribución

objetivo actualizada q_4 222D) es aceptado (p. ej., basado en criterios de aceptación U_4 224D) y emitido como el token seleccionado Y 230.

El ejemplo 200B ilustrado en **la Figura 2B** ilustra un ejemplo en el que el modelo objetivo rechaza cada uno de los tokens 220A-220D generados por el modelo de borrador. En este ejemplo, el modelo objetivo puede generar una distribución objetivo final 222E, muestrear un token de esta distribución objetivo final 222E y terminar el recorrido del árbol generado. Como se ilustra, la distribución objetivo final q_5 222E puede ser el resultado de restar la probabilidad asociada con el token final en el conjunto de tokens generados (p. ej., la probabilidad p_4 asociada con el token X_4 220D). La distribución objetivo final q_5 222E puede estar representada por la ecuación $q_5 = (q_4 - p_4)^+$. Un token 240 muestreado de la distribución objetivo final q_5 222E puede devolverse como la salida Y de la realización de un muestreo de rechazo recursivo en los tokens generados.

La Figura 3 ilustra un ejemplo de estructura de datos de árbol 300 de tokens generados usando decodificación especulativa recursiva en modelos de inteligencia artificial generativa, de acuerdo con aspectos de la presente divulgación.

Como se ilustra, la estructura de datos de árbol 300 incluye una pluralidad de niveles, comenzando desde un

nodo raíz 310 que representa la entrada (que puede tokenizarse) en un modelo de inteligencia artificial generativa y tokens previamente generados por el modelo de borrador y aceptados por el modelo objetivo usando muestreo de rechazo recursivo, como se analizó anteriormente. Cada nivel 320, 330 y 340 (y/u otros no ilustrados en la **Figura 3**) en la estructura de datos de árbol 300 puede ser el resultado de generar especulativamente un número de tokens por el modelo de borrador, dado algún conjunto previo de tokens generados. El número de tokens generados en el nivel $n+1$ de la estructura de datos de árbol 300 para un nodo en el nivel n de la estructura de datos de árbol 300 puede determinarse basado en un factor de ramificación definido para nodos en el nivel n de la estructura de datos de árbol 300. Como se analizó, el factor de ramificación puede ser el mismo en todos los niveles de la estructura de datos de árbol 300, o el factor de ramificación puede definirse para cada nivel de la estructura de datos de árbol 300. En algunos aspectos, el factor de ramificación puede definirse con un mayor número de ramas generadas para nodos en niveles más bajos de la estructura de datos de árbol 300 y con un menor número de ramificaciones para nodos en niveles superiores de la estructura de datos de árbol 300, o viceversa.

Para atravesar la estructura de datos de árbol 300 y seleccionar un conjunto de tokens de grupos de tokens generados especulativamente, el modelo objetivo puede

aceptar o rechazar de forma recursiva tokens a partir del primer conjunto de tokens generados por el modelo de borrador en el primer nivel 320 de la estructura de datos de árbol 300 (p. ej., tokens representados por nodos que son los nodos hijos inmediatos del nodo raíz 310). Generalmente, al atravesar el árbol, el modelo objetivo puede determinar si un token debe ser aceptado o rechazado y ajustar la distribución objetivo basado en determinar si el token debe aceptarse o rechazarse. En el primer nivel 320 de la estructura de datos de árbol 300, si se rechaza un primer token, entonces el modelo objetivo puede eliminar el primer token de la distribución objetivo (como se analizó anteriormente con respecto a la **Figura 2**) y proceder a determinar si un segundo token en el primer nivel del árbol debe aceptarse o rechazarse basado en una distribución objetivo actualizada. Al rechazar el primer token en el primer nivel 320 de la estructura de datos de árbol 300, el modelo objetivo no necesita analizar los tokens representados por nodos que son, directa o indirectamente, nodos secundarios del nodo en la estructura de datos de árbol 300 que representan el primer token rechazado.

Generalmente, al atravesar la estructura de datos de árbol 300, la distribución objetivo puede modificarse recursivamente hasta que se alcance una condición de terminación. Una condición de terminación puede ser, por ejemplo, la identificación de una secuencia de tokens que se

aceptan como salidas de respuesta aceptables para la entrada (representada por el nodo raíz 310 de la estructura de datos de árbol 300), más un nodo adicional muestreado de la distribución objetivo modificada después de que la secuencia de tokens haya sido aceptada por el modelo objetivo. En otro ejemplo, una condición de terminación puede ser una determinación de que el modelo objetivo no ha aceptado ninguna secuencia de tokens generados especulativamente por el modelo de borrador. En tal caso, como se analizó anteriormente con respecto a la **Figura 2B**, un token adicional muestreado de la distribución objetivo final puede emitirse como una respuesta a la entrada correspondiente al nodo raíz 310 en la estructura de datos de árbol 300. La distribución objetivo, como se analizó anteriormente, puede ser una distribución objetivo generada al eliminar las probabilidades asociadas con los tokens generados de la distribución objetivo original generada por el modelo de borrador.

En el ejemplo ilustrado en la **Figura 3**, en el primer nivel 320 de la estructura de datos de árbol 300, puede rechazarse un primer token 322A. Debido a que el primer token 322A se rechaza como uno de los tokens candidatos para su inclusión en una respuesta a la entrada correspondiente al nodo raíz 310, los tokens en la estructura de datos de árbol 300 que son tokens hijos del primer token 322A no necesitan ser examinados para su selección como tokens candidatos. Por lo tanto, los tokens hijos inmediatos al

primer token 322A en el segundo nivel 330 de la estructura de datos de árbol 300 y los tokens de nietos al primer token 322A en el tercer nivel 340 de la estructura de datos de árbol 300 pueden descartarse o ignorarse.

5 Al rechazar el primer token 322A del conjunto de tokens candidatos en el primer nivel 320 de la estructura de datos de árbol 300, como se ha analizado, la distribución de probabilidad objetivo asociada con los tokens en el primer nivel 320 de la estructura de datos de árbol 300
10 puede modificarse para eliminar el valor de probabilidad asociado con el primer token 322A. La distribución objetivo actualizada puede usarse posteriormente para examinar un segundo token 322B en el primer nivel 320 de la estructura de datos de árbol 300. Como se ilustra, el segundo token
15 322B puede aceptarse como un token candidato para su inclusión en una respuesta a la entrada correspondiente al nodo raíz 310, y un análisis puede proceder a los tokens hijos (p. ej., 332A, 332B y 332C) del segundo token 322B en el segundo nivel 330 de la estructura de datos de árbol 300.

20 Como se ilustra, los tokens 332A y 332B pueden ser rechazados por el modelo objetivo. Al rechazar los tokens 332A y 332B, la distribución de probabilidad objetivo usada para aceptar o rechazar tokens en el segundo nivel 330 de la estructura de datos de árbol 300 puede ajustarse para
25 eliminar las probabilidades asociadas con los tokens 332A y 332B. Esta distribución de probabilidad objetivo ajustada

puede usarse entonces para determinar si aceptar o rechazar el token 332C en el segundo nivel 330 de la estructura de datos de árbol 300. Como se ilustra, el modelo objetivo puede aceptar el token 332C, y un análisis puede proceder a los tokens hijos (p. ej., 342A, 342B) del token 332C en el tercer nivel 340 de la estructura de datos de árbol 300. En el tercer nivel 340 de la estructura de datos de árbol 300, el modelo objetivo puede examinar los tokens 342A y 342B usando las técnicas analizadas anteriormente y puede aceptar el token 342A como un candidato para su inclusión en una respuesta a la entrada correspondiente al nodo raíz 310.

En algunos aspectos, la aceptación del segundo token 322B como un token candidato para su inclusión en una respuesta a la entrada correspondiente al nodo raíz 310 puede no impedir el análisis del tercer token 322C para su inclusión o rechazo como un token candidato para su inclusión en una respuesta a la entrada correspondiente al nodo raíz 310. En tal caso, la aceptación o rechazo de los tokens hijos en los niveles 320 y 330 de la estructura de datos de árbol 300 (entre otros, no ilustrados en la **Figura 3**) puede realizarse como se analizó anteriormente con respecto a la aceptación o rechazo de tokens hijos del segundo token 322B anterior. En algunos aspectos, tales como cuando la probabilidad de que el segundo token 322B sea un token candidato para una respuesta a la entrada correspondiente al nodo raíz 310 excede un valor umbral, puede considerarse que el tercer

token 322C ha sido rechazado, y un análisis de los tokens hijos del tercer token 322C puede omitirse.

Las técnicas de decodificación especulativa recursiva analizadas en la presente pueden conducir a ganancias significativas en el rendimiento de generación de tokens en relación con la generación y aceptación por token. Generalmente, las técnicas de decodificación especulativa recursiva pueden permitir aumentos en el rendimiento de aceptación de tokens (p. ej., el número de tokens que se aceptan para una longitud de borrador dada) para cada vez que se ejecuta el modelo objetivo en relación con las técnicas de decodificación especulativa por token. La cantidad en la que aumenta el rendimiento puede expandirse a medida que aumenta el número de ramas generadas para cualquier token y a medida que aumenta la duración del borrador. Además, las técnicas de decodificación especulativa recursiva analizadas en la presente pueden proporcionar tasas aumentadas (a lo largo del tiempo) a las que se generan tokens en relación con las técnicas de decodificación especulativa por token.

20

Ejemplo de operaciones para decodificación especulativa recursiva en modelos de inteligencia artificial generativa

La Figura 4 ilustra operaciones de ejemplo 400 que pueden ser realizadas por un dispositivo informático para generar una respuesta a un indicador de entrada usando modelos de inteligencia artificial generativa (p. ej.,

25

como se analiza en la presente con respecto a las **Figuras 1 a 3**), de acuerdo con aspectos de la presente divulgación. Las operaciones 400 pueden ser realizadas por un dispositivo informático en el que se puede implementar al menos un modelo objetivo, tal como un computador portátil, un computador de escritorio, un servidor, una instancia informática en la nube alojada en un entorno informático distribuido, o lo similar.

Como se ilustra, las operaciones 400 comienzan en el bloque 410, recibiendo una pluralidad de conjuntos de tokens. Generalmente, la pluralidad de conjuntos de tokens pueden ser tokens generados (p. ej., mediante un modelo de borrador desplegado en un dispositivo de cliente desde el que se recibe la pluralidad de conjuntos de tokens) basado en un indicador de entrada y un primer modelo de inteligencia artificial generativa. El mensaje de entrada puede recibirse en el dispositivo de cliente, por ejemplo, de un usuario a través de un mensaje de entrada de texto, indicaciones de captura de audio u otras técnicas mediante las cuales se puede proporcionar una entrada al dispositivo cliente. Generalmente, cada conjunto de tokens en la pluralidad de conjuntos de tokens puede corresponder a una respuesta candidata al indicador de entrada.

En algunos aspectos, la pluralidad de conjuntos de tokens puede organizarse en una estructura de datos de árbol (p. ej., la estructura de datos de árbol 110 ilustrada

en la **Figura 1** o la estructura de datos de árbol 300
ilustrada en la **Figura 3**). La estructura de datos de árbol
puede tener un nodo raíz y uno o más nodos de hoja. El
indicador de entrada puede corresponder al nodo raíz de la
5 estructura de datos de árbol. Cada ruta a través de la
estructura de datos de árbol puede corresponder a una
respuesta candidata diferente al indicador de entrada. Debe
entenderse que una ruta a través de la estructura de datos
de árbol no necesita terminar en un nodo de la estructura
10 de datos de árbol que no tiene nodos de hoja; es decir, una
respuesta candidata puede corresponder a una ruta que
atraviesa parcialmente la estructura de datos de árbol (p.
ej., para reflejar tokens rechazados en una respuesta).

En algunos aspectos, el número de tokens en
15 cualquier nivel dado de la estructura de datos de árbol puede
estar basado en un factor de ramificación asociado con un
nivel inmediatamente anterior al nivel en la estructura de
datos de árbol. La estructura de datos de árbol puede tener
diferentes factores de ramificación asignados a diferentes
20 niveles en la estructura de datos de árbol. Generalmente,
los factores de ramificación más altos pueden permitir la
generación de un mayor número de tokens en cualquier nivel
dado de la estructura de datos de árbol y generalmente
aumentar el número total de tokens incluidos en la estructura
25 de datos de árbol. Mientras tanto, los factores de
ramificación más bajos pueden restringir la generación de

tokens en cualquier nivel dado de la estructura de datos de árbol y generalmente disminuyen el número total de tokens incluidos en la estructura de datos de árbol.

En algunos aspectos, el tamaño de cada conjunto de tokens puede basarse en una métrica de complejidad computacional asociada con generar un conjunto de tokens objetivo por el segundo modelo de inteligencia artificial generativa.

En el bloque 420, las operaciones 400 proceden a seleccionar, usando un segundo modelo de inteligencia artificial generativa y ajuste recursivo de una distribución de probabilidad, un conjunto de tokens de la pluralidad de conjuntos de tokens.

En algunos aspectos, el ajuste recursivo de la distribución de probabilidad incluye rechazar un token de la pluralidad de conjuntos de tokens como un candidato para su inclusión como respuesta a un indicador de entrada. Generalmente, un token puede rechazarse si no se cumplen algunos criterios de aceptación definidos, que pueden basarse en una distribución objetivo y una probabilidad asociada con el token en la distribución objetivo original. La distribución actual usada por el modelo objetivo para determinar si un token debe ser aceptado o rechazado puede modificarse al eliminar (o restar) la probabilidad asociada con el token en la distribución objetivo original de la distribución actual, lo que da como resultado la

generación de una distribución objetivo actualizada que se puede usar para determinar si el siguiente token debe aceptarse o rechazarse como un candidato para su inclusión como respuesta al indicador de entrada.

5 En algunos aspectos, seleccionar el conjunto de tokens de la pluralidad de conjuntos de tokens puede incluir rechazar un primer token en un primer nivel de una estructura de datos de árbol que representa la pluralidad de conjuntos de tokens. Se puede generar una distribución
10 de probabilidad ajustada basada en el rechazo del primer token. Dentro de la estructura de datos de árbol, los tokens hijos del primer token en niveles más profundos que el primer nivel de la estructura de datos de árbol pueden descartarse. Un segundo token en el primer nivel de la
15 estructura de datos de árbol puede aceptarse o rechazarse basado en la distribución de probabilidad ajustada.

 En algunos aspectos, seleccionar el conjunto de tokens de la pluralidad de conjuntos de tokens puede incluir rechazar cada conjunto de tokens generado por el
20 primer modelo de inteligencia artificial generativa. Se puede muestrear un token, usando el segundo modelo de inteligencia artificial generativa, basado en una distribución objetivo que excluye las probabilidades asociadas con cada conjunto de tokens generado por el
25 primer modelo de inteligencia artificial generativa.

En el bloque 430, las operaciones 400 proceden con emitir el conjunto de tokens seleccionado como una respuesta al indicador de entrada.

En algunos aspectos, el primer modelo de
5 inteligencia artificial generativa puede corresponder a un modelo de borrador en una canalización de decodificación especulativa, y el segundo modelo de inteligencia artificial generativa puede corresponder a un modelo objetivo en la canalización de decodificación especulativa.
10 El primer modelo de inteligencia artificial generativa puede ejecutarse en un dispositivo cliente, y el segundo modelo de inteligencia artificial generativa puede ejecutarse en un sistema remoto desde el dispositivo de cliente, tal como un computador servidor, una instancia de
15 computación en la nube, o lo similar. En algunos aspectos, el primer modelo de inteligencia artificial generativa y el segundo modelo de inteligencia artificial generativa pueden ejecutarse en el mismo dispositivo.

En algunos aspectos, el primer modelo de
20 inteligencia artificial generativa y el segundo modelo de inteligencia artificial generativa pueden tener distribuciones de probabilidad equivalentes.

En algunos aspectos, el primer modelo de
inteligencia artificial generativa puede tener una
25 distribución de probabilidad que se aproxime a una

distribución de probabilidad asociada al segundo modelo de inteligencia artificial generativa.

Ejemplo de decodificación autoespeculativa en modelos de 5 inteligencia artificial generativa

En algunos aspectos, diversos tipos de decodificación especulativa, incluida la decodificación especulativa grupal y la decodificación especulativa recursiva, pueden lograrse usando un único modelo de
10 inteligencia artificial generativa que combine la funcionalidad del modelo de borrador y el modelo objetivo analizado anteriormente. Al hacerlo, la generación de tokens en borrador, la generación de tokens objetivo y la aceptación de tokens pueden paralelizarse en un único modelo de
15 inteligencia artificial generativa. El uso de un único modelo de inteligencia artificial generativa puede, por ejemplo, reducir el gasto computacional que implica generar tanto un modelo objetivo como un modelo de borrador, aumentar el rendimiento de las tareas generativas mediante la ejecución
20 de la verificación de tokens y la generación especulativa en un único pase a través del modelo de inteligencia artificial generativa único, reducir la cantidad de memoria usada en el almacenamiento de modelos usados para la decodificación especulativa en tareas generativas, y así sucesivamente.

25 **La Figura 5** ilustra un canalización de ejemplo
500 para decodificación autoespeculativa en modelos de

inteligencia artificial generativa, de acuerdo con aspectos de la presente divulgación. En algunos aspectos, la canalización 500 puede usarse para generar y analizar una estructura de datos de árbol de tokens asociados con una respuesta candidata a una entrada de consulta en un modelo de inteligencia generativa, tal como la estructura de datos de árbol 300 ilustrada en la **Figura 3 o lo similar**.

Como se ilustra, la canalización 500 usa un único modelo de inteligencia artificial generativa para generar especulativamente tokens y verificar los tokens generados especulativamente. Durante una primera ronda de inferencia en la canalización 500, se genera especulativamente un primer conjunto de tokens 502. Como se ilustra, por ejemplo, el primer conjunto de tokens 502 puede incluir tokens 1 a 4 y puede proporcionarse como entrada durante una segunda ronda en la canalización 500 para generar especulativamente el siguiente conjunto de tokens como un proceso por lotes en el que se generan múltiples conjuntos de tokens. Si bien el primer conjunto de tokens generados especulativamente es procesado por el modelo de inteligencia artificial generativa única, el modelo de inteligencia artificial generativa única continúa generando especulativamente una pluralidad de segundos conjuntos de tokens de borrador 504, 506, 508 y 510 en una segunda ronda de inferencia en la tubería 500.

Al generar los segundos conjuntos de tokens de borrador 504, 506, 508 y 510, se pueden hacer suposiciones

para diferentes números de tokens aceptados del primer conjunto de tokens 502. Por ejemplo, como se ilustra, el segundo conjunto de tokens de borrador 504 puede asumir la aceptación del primer token de borrador del primer conjunto de tokens 502 y puede incluir un conjunto de tokens generado especulativamente basado en la aceptación del primer token. El segundo conjunto de tokens de borrador 506 puede asumir la aceptación del primer y segundo tokens de borrador del primer conjunto de tokens 502 e incluir un conjunto de tokens generado especulativamente basado en la aceptación del primer y segundo tokens. El segundo conjunto de tokens de borrador 508 puede asumir la aceptación del primer a tercer tokens de borrador del primer conjunto de tokens 502 e incluir un conjunto de tokens generado especulativamente basado en la aceptación del primer a tercer tokens. Finalmente, el segundo conjunto de tokens de borrador 510 puede asumir la aceptación de los cuatro tokens del primer conjunto de tokens 502 e incluir un conjunto de tokens generado especulativamente basado en la aceptación de los cuatro tokens. En diversos aspectos, para los casos en los que se supone que se aceptan menos tokens que el número de tokens incluidos en el primer conjunto de tokens 502, se puede añadir protección 503 (p. ej., valores nulos, constantes predefinidas, etc.) de manera que cada suposición sea de la misma longitud.

Una vez que el único modelo de inteligencia artificial generativa completa el muestreo de rechazo en el conjunto de tokens generado especulativamente, el modelo de inteligencia artificial generativa única selecciona el conjunto de tokens generados especulativamente asociados con el conjunto de tokens aceptados del primer conjunto como entrada al único modelo de inteligencia artificial generativa para otra ronda de inferencia en la que los tokens se generan especulativamente usando el modelo de inteligencia artificial generativa única. En este ejemplo, se puede ver que los cuatro tokens en el primer conjunto de tokens 502 han sido aceptados por el modelo de inteligencia artificial generativa única como una verificación de borrador 512, y por lo tanto, el conjunto de tokens 510 puede usarse para una generación especulativa adicional de tokens usando el modelo de inteligencia artificial generativa única.

El proceso anterior puede continuar hasta que se produzca un evento de terminación. Las rondas sucesivas de generación especulativa pueden basarse en supuestos de que el número de tokens de una ronda previa de generación especulativa sea aceptado por el único modelo de inteligencia artificial generativa. Por ejemplo, como se ilustra en la Figura 5, se pueden generar conjuntos de tokens de borrador 522, 524, 526 y 528 en la ronda $k + 1$ de inferencias con los tokens incluidos en los conjuntos de tokens de borrador que se basan en un número de tokens

generados especulativamente más allá de las N tokens aceptados generados en la ronda $k - 1$ de inferencias. En este ejemplo, se puede ver que los cuatro tokens generados especulativamente generados durante la ronda k de inferencias han sido aceptadas como un verificación de borrador 520, y los tokens $N + 5$ a $N + 8$ pueden usarse para una mayor generación especulativa de tokens usando el único modelo de inteligencia artificial generativa.

En algunos aspectos, un evento de terminación puede incluir la generación de un token especial usado para denotar el final de una respuesta (p. ej., que no se puedan incluir más tokens plausiblemente en una respuesta debido a las probabilidades asociadas con estos tokens que caigan por debajo de un valor umbral de probabilidad de aceptación). Un evento de terminación puede, en algunos aspectos, alcanzarse cuando se ha generado un número umbral de tokens.

En algunos aspectos, cuando todos los tokens de una ronda anterior de generación de tokens especulativos son rechazados por el único modelo de inteligencia artificial generativa, el proceso puede reiniciarse con el último conjunto de tokens aceptados, más un token muestreado de una distribución final (p. ej., como se analizó anteriormente), proporcionándose como entrada al modelo de inteligencia artificial generativa única.

La Figura 6 ilustra arquitecturas de ejemplo 600A, 600B para decodificación autoespeculativa en modelos

de inteligencia artificial generativa, de acuerdo con aspectos de la presente divulgación. Las arquitecturas de ejemplo 600A y 600B pueden permitir la generación de múltiples tokens en cualquier pase a través del modelo, tal como en la generación de tokens ilustrada en **las Figuras 1, 2A, 2B, 3, 4 y 5**, como se analizó anteriormente.

En la arquitectura de ejemplo 600A, un modelo de inteligencia artificial generativa 610 puede entrenarse para generar múltiples incrustaciones de indicadores de pronósticos 612, anexados a un conjunto de tokens de entrada, para permitir la generación paralela de múltiples tokens de salida 614. Estas incrustaciones de indicadores de pronósticos pueden ser incrustaciones que corresponden a tokens que se incluyen en una respuesta a un indicador de entrada (incluido cualquier token previamente generado y aceptado). El modelo de inteligencia artificial generativa 610 puede ser un modelo de inteligencia artificial generativa, tal como un modelo de lenguaje grande preentrenado u otro modelo de inteligencia artificial generativa preentrenado, actualizado usando diversas técnicas de ajuste fino. Por ejemplo, un modelo de inteligencia artificial generativa usado para generar respuestas textuales a entradas textuales (también conocido como un modelo de lenguaje grande) puede actualizarse usando técnicas tal como la adaptación de bajo rango (LoRA) de modelos de lenguaje grande.

En la arquitectura de ejemplo 600B, los modelos de inteligencia artificial generativa pueden implementarse como un modelo autorregresivo parcial. Las operaciones de inferencia, usadas para generar tokens especulativamente, pueden realizarse usando un subconjunto de capas en el modelo autorregresivo parcial (p. ej., las n capas superiores del modelo o las n capas inferiores del modelo). Al hacerlo, las capas usadas para generar tokens especulativamente pueden crear un contexto que puede permitir modelar la causalidad y/u otras relaciones para los tokens generados especulativamente que pueden ser alimentados como entrada en la porción del modelo que verifica los tokens como respuestas válidas al indicador de entrada.

La arquitectura 600B puede implementarse de diversas maneras de tal manera que la inferencia autorregresiva, y la generación de múltiples conjuntos de tokens para aceptación y/o rechazo, pueden generarse usando un pequeño número de capas autorregresivas en un modelo de inteligencia artificial generativa. En la implementación de ejemplo 620, un modelo de inteligencia artificial generativa puede incluir una pluralidad de capas no autorregresivas 622A-622C y una capa autorregresiva 624. Las capas en el modelo de inteligencia artificial generativa pueden estar organizadas en una pila, con la capa más baja de la pila correspondiente a la capa que recibe una entrada para su procesamiento y la capa más alta

en la pila correspondiente a la capa que genera una salida. En la implementación 620, las capas no autorregresivas 622A-622C pueden colocarse en la parte inferior de la pila, y la capa autorregresiva 624 puede colocarse en la parte superior de la pila. En contraste, en la implementación de ejemplo 630, las capas del modelo de inteligencia artificial generativa pueden organizarse de tal manera que una capa autorregresiva 632 se coloca en la parte inferior de la pila, y las capas no autorregresivas 632A-632C se colocan en la parte superior de la pila. Estas capas autorregresivas 624 y 632 pueden operar, por ejemplo, en un bucle para generar y aceptar continuamente tokens a emitirse como una respuesta a un indicador de entrada (y, en algunos aspectos, tokens generados previamente incluidos como una respuesta parcial al indicador de entrada).

Ejemplo de operaciones para generar respuestas a consultas de entrada mediante decodificación autoespeculativa en modelos de inteligencia artificial generativa

La Figura 7 ilustra operaciones de ejemplo 700 que pueden ser realizadas por un dispositivo informático para generar una respuesta a un indicador de entrada usando modelos de inteligencia artificial generativa, de acuerdo con aspectos de la presente divulgación. Las operaciones 700 pueden ser realizadas por un dispositivo en el que se puede desplegar al menos un modelo de inteligencia

artificial generativa que puede actuar tanto como un modelo de borrador como un modelo objetivo, tal como un teléfono inteligente, una tableta, un computador portátil, un computador de escritorio, un servidor, una instancia de computación en la nube alojada en un entorno informático distribuido, o lo similar.

Como se ilustra, las operaciones 700 comienzan en el bloque 710, generando, basado en un indicador de entrada y un modelo de inteligencia artificial generativa, una primera pluralidad de conjuntos de tokens. En algunos aspectos, los conjuntos de tokens pueden incluir secuencias de tokens que pueden aceptarse, en parte o en su totalidad, como una respuesta candidata al indicador de entrada.

En algunos aspectos, la primera pluralidad de conjuntos de tokens puede representarse como una estructura de datos de árbol. Dentro de esta estructura de datos de árbol, el indicador de entrada puede corresponder a un nodo raíz. Cada conjunto de tokens en la primera pluralidad de conjuntos de tokens puede representarse mediante una ruta navegable a través de la estructura de datos de árbol (p. ej., como se analizó anteriormente con respecto a la **Figura 3**).

En algunos aspectos, la profundidad de la estructura de datos de árbol puede corresponder a un número máximo de tokens generados por un único pase a través del modelo de inteligencia artificial generativa.

En algunos aspectos, la profundidad de la estructura de datos de árbol puede corresponder a un número máximo de tokens generados por un único pase a través del modelo de inteligencia artificial generativa.

5 En algunos aspectos, el tamaño máximo de la estructura de datos de árbol se puede establecer basado en una métrica de complejidad computacional asociada con generar un conjunto de tokens por el modelo de inteligencia artificial generativa.

10 En el bloque 720, las operaciones 700 proceden a generar especulativamente, usando el modelo de inteligencia artificial generativa, una segunda pluralidad de conjuntos de tokens.

 En el bloque 730, las operaciones 700 proceden
15 con la selección, mientras que se genera especulativamente la segunda pluralidad de conjuntos de tokens, un conjunto de tokens de la primera pluralidad de conjuntos de tokens.

 En algunos aspectos, seleccionar el conjunto de tokens de la primera pluralidad de conjuntos de tokens puede
20 incluir seleccionar una secuencia más larga de tokens aceptados de la primera pluralidad de conjuntos de tokens.

 En algunos aspectos, los conjuntos de tokens en la segunda pluralidad de conjuntos de tokens pueden incluir protección responsable de un número de tokens en el conjunto
25 de tokens seleccionado de la primera pluralidad de tokens. Por ejemplo, para un número de tokens N incluidos en cada

conjunto de tokens de la primera pluralidad de conjuntos de tokens, el número seleccionado de tokens puede incluir entre 0 y N tokens. En tal caso, la protección incluida en los conjuntos de tokens en la segunda pluralidad de conjuntos de tokens puede incluir ningún relleno (correspondiente a la aceptación de todos los N tokens en un conjunto de tokens de la primera pluralidad de tokens), un token de relleno (correspondiente a la aceptación de $N-1$ tokens), dos tokens de relleno (correspondientes a la aceptación de $N-2$ tokens), y así sucesivamente.

En algunos aspectos, para seleccionar el conjunto de tokens de la primera pluralidad de tokens, un primer token puede rechazarse en un primer nivel de la estructura de datos de árbol. Basado en el rechazo del primer token, se puede generar una distribución de probabilidad ajustada, y la estructura de datos de árbol se puede podar al descartar tokens hijos del primer token de la estructura de datos de árbol. Los tokens hijos del primer token pueden descartarse porque el rechazo del primer token generalmente implica que los tokens hijos del primer token no pueden ser respuestas válidas al mensaje de entrada. Después se puede determinar si aceptar o rechazar un segundo token en el primer nivel de la estructura de datos de árbol basado en la distribución de probabilidad ajustada.

En algunos aspectos, seleccionar el conjunto de tokens de la primera pluralidad de conjuntos de tokens puede incluir rechazar cada conjunto de tokens en la primera pluralidad de conjuntos de tokens generados por el
5 modelo de inteligencia artificial generativa. Un token puede muestrearse usando el modelo de inteligencia artificial generativa basado en una distribución objetivo que excluye las probabilidades asociadas con cada conjunto de tokens en la primera pluralidad de conjuntos de tokens.
10 El conjunto de tokens seleccionado de la primera pluralidad de conjuntos de tokens puede incluir el token muestreado.

En el bloque 740, las operaciones 700 proceden con emitir el conjunto de tokens seleccionado de la primera pluralidad de tokens y un conjunto de tokens asociado en
15 la segunda pluralidad de conjuntos de tokens como una respuesta al indicador de entrada.

En algunos aspectos, el modelo de inteligencia artificial generativa comprende un modelo de inteligencia artificial generativa entrenado para generar múltiples
20 tokens en respuesta al indicador de entrada basado en incrustaciones de indicadores de pronósticos.

En algunos aspectos, el modelo de inteligencia artificial generativa comprende un modelo que incluye una o más capas no autorregresivas y una o más capas
25 autorregresivas. En algunos aspectos, la una o más capas autorregresivas pueden incluir una o más capas en una parte

superior de una pila de capas que representan el modelo de inteligencia artificial generativa. En algunos aspectos, la una o más capas autorregresivas pueden incluir una o más capas en la parte inferior de una pila de capas que representan el modelo de inteligencia artificial generativa.

Ejemplos de sistemas de procesamiento para decodificación especulativa recursiva en modelos de inteligencia artificial generativa

La Figura 8 representa un sistema de procesamiento de ejemplo 800 para generar una respuesta a una entrada de consulta en un modelo de inteligencia artificial generativa basado en decodificación especulativa recursiva, tal como se divulgan en la presente, por ejemplo, con respecto a la Figura 4.

El sistema de procesamiento 800 incluye una unidad de procesamiento central (CPU, en inglés) 802, que en algunos ejemplos puede ser una CPU multinúcleo. Las instrucciones ejecutadas en la CPU 802 pueden cargarse, por ejemplo, desde una memoria de programa asociada con la CPU 802 o pueden cargarse desde una partición de memoria (p. ej., de una memoria 824).

El sistema de procesamiento 800 también incluye componentes de procesamiento adicionales adaptados a funciones específicas, tales como una unidad de procesamiento de gráficos (GPU, en inglés) 804, un

procesador de señal digital (DSP, en inglés) 806, una unidad de procesamiento neuronal (NPU, en inglés) 808 y un componente de conectividad 812.

Una NPU, tal como la NPU 808, es generalmente un
5 circuito especializado configurado para implementar lógica de control y aritmética para ejecutar algoritmos de aprendizaje automático, tales como algoritmos para procesar redes neuronales artificiales (ANN, en inglés),
redes neuronales profundas (DNN, en inglés), bosques
10 aleatorios (RF, en inglés), y lo similar. A veces, una NPU puede denominarse alternativamente procesador de señal neuronal (NSP, en inglés), unidad de procesamiento de tensores (TPU, en inglés), procesador de red neuronal (NNP, en inglés), unidad de procesamiento de inteligencia (IPU,
15 en inglés), unidad de procesamiento de visión (VPU, en inglés) o unidad de procesamiento de gráficos.

Las NPU, como la NPU 808, se configuran para acelerar el rendimiento de tareas comunes de aprendizaje automático, como clasificación de imágenes, traducción
20 automática, detección de objetos y varios otros modelos predictivos. En algunos ejemplos, se puede instanciar una pluralidad de NPU en un único chip, tal como un sistema en un chip (SoC, en inglés), mientras que en otros ejemplos, tales NPU pueden ser parte de un acelerador de red
25 neuronal dedicado.

Las NPU pueden estar optimizadas para capacitación o inferencia, o en algunos casos configuradas para equilibrar el rendimiento entre ambas. Para las NPU que son capaces de realizar tanto el entrenamiento como la
5 inferencia, las dos tareas aún pueden realizarse generalmente de manera independiente.

Las NPU diseñadas para acelerar el entrenamiento generalmente se configuran para acelerar la optimización de nuevos modelos, que es una operación altamente intensiva
10 en computación que implica introducir un conjunto de datos existente (a menudo marcado o etiquetado), iterar sobre el conjunto de datos y después ajustar los parámetros del modelo, tales como pesos y sesgos, para mejorar el rendimiento del modelo. Generalmente, la optimización
15 basada en una predicción incorrecta implica propagarse de nuevo a través de las capas del modelo y determinar gradientes para reducir el error de predicción.

Las NPU diseñadas para acelerar la inferencia generalmente se configuran para operar en modelos
20 completos. Por lo tanto, tales NPU pueden configurarse para ingresar una nueva pieza de datos y procesar rápidamente esta nueva pieza a través de un modelo ya entrenado para generar una salida de modelo (p. ej., una inferencia).

En algunas implementaciones, la NPU 808 es una
25 parte de una o más de la CPU 802, la GPU 804 y/o el DSP 806. Estos pueden estar ubicados en un equipo de usuario

(UE, en inglés) en un sistema de comunicación inalámbrica u otro dispositivo informático.

En algunos ejemplos, el componente de conectividad 812 puede incluir subcomponentes, por ejemplo, para conectividad de tercera generación (3G), conectividad de cuarta generación (4G) (p. ej., Evolución a Largo Plazo (LTE, en inglés)), conectividad de quinta generación (5G) (p. ej., Nueva Radio (NR, en inglés)), conectividad Wi-Fi, conectividad Bluetooth y otros estándares de transmisión inalámbrica de datos. El componente de conectividad 812 puede además acoplarse a una o más antenas 814.

El sistema de procesamiento 800 también puede incluir una o más unidades de procesamiento de sensores 816 asociadas con cualquier manera de sensor, uno o más procesadores de señal de imagen (ISP, en inglés) 818 asociados con cualquier manera de sensor de imagen, y/o un procesador de navegación 820, que puede incluir componentes del sistema de posicionamiento basado en satélite (p. ej., GPS o GLONASS) así como componentes del sistema de posicionamiento inercial.

El sistema de procesamiento 800 también puede incluir uno o más dispositivos de entrada y/o salida 822, tales como pantallas, superficies sensibles al tacto (incluidas pantallas sensibles al tacto), botones físicos, altavoces, micrófonos y lo similar.

En algunos ejemplos, uno o más de los procesadores del sistema de procesamiento 800 pueden basarse en un conjunto de instrucciones ARM o RISC-V.

El sistema de procesamiento 800 también incluye
5 la memoria 824, que es representativa de una o más memorias estáticas y/o dinámicas, tales como una memoria dinámica de acceso aleatorio, una memoria estática basada en flash, y lo similar. En este ejemplo, la memoria 824 incluye componentes ejecutables por computador, que pueden
10 ejecutarse por uno o más de los procesadores antes mencionados del sistema de procesamiento 800.

En particular, en este ejemplo, la memoria 824 incluye un componente receptor de conjunto de tokens 824A, un componente de selección de token 824B, un componente de
15 generación de salida 824C y modelos de inteligencia artificial generativa 824D. Los componentes representados, y otros no representados, pueden configurarse para realizar diversos aspectos de los métodos divulgados en la presente.

Generalmente, el sistema de procesamiento 800
20 y/o componentes de este pueden configurarse para realizar los métodos divulgados en la presente.

Ejemplos de sistemas de procesamiento para decodificación
autoespeculativa en modelos de inteligencia artificial
generativa

La **Figura 9** representa un sistema de
5 procesamiento de ejemplo 900 para generar una respuesta a
una entrada de consulta en un modelo de inteligencia
artificial generativa basado en la decodificación
autoespeculativa, tal como se divulgan en la presente, por
ejemplo, con respecto a **la Figura 7**.

10 El sistema de procesamiento 900 incluye una unidad
de procesamiento central (CPU) 902, que en algunos ejemplos
puede ser una CPU multinúcleo. Las instrucciones ejecutadas
en la CPU 902 pueden cargarse, por ejemplo, desde una memoria
de programa asociada con la CPU 902 o pueden cargarse desde
15 una partición de memoria (p. ej., de una memoria 924).

El sistema de procesamiento 900 también incluye
componentes de procesamiento adicionales adaptados a
funciones específicas, tales como una unidad de
procesamiento de gráficos (GPU) 904, un procesador de señal
20 digital (DSP) 906, una unidad de procesamiento neuronal
(NPU) 908 y un componente de conectividad 912.

Una NPU, tal como la NPU 908, es generalmente un
circuito especializado configurado para implementar control
y lógica aritmética para ejecutar algoritmos de aprendizaje
25 automático, tales como algoritmos para procesar redes
neuronales artificiales (ANN), redes neuronales profundas

(DNN), bosques aleatorios (RF), y lo similar. A veces, una NPU puede denominarse alternativamente procesador de señal neuronal (NSP, en inglés), unidad de procesamiento de tensores (TPU, en inglés), procesador de red neuronal (NNP, en inglés), unidad de procesamiento de inteligencia (IPU, en inglés), unidad de procesamiento de visión (VPU, en inglés) o unidad de procesamiento de gráficos.

Las NPU, como la NPU 908, están configuradas para acelerar el rendimiento de tareas comunes de aprendizaje automático, tales como clasificación de imágenes, traducción automática, detección de objetos y varios otros modelos predictivos. En algunos ejemplos, se puede instanciar una pluralidad de NPU en un único chip, tal como un sistema en un chip (SoC), mientras que en otros ejemplos tales NPU pueden ser parte de un acelerador de red neuronal dedicado.

Las NPU pueden estar optimizadas para capacitación o inferencia, o en algunos casos configuradas para equilibrar el rendimiento entre ambas. Para las NPU que son capaces de realizar tanto el entrenamiento como la inferencia, las dos tareas aún pueden realizarse generalmente de manera independiente.

Las NPU diseñadas para acelerar el entrenamiento generalmente se configuran para acelerar la optimización de nuevos modelos, que es una operación altamente intensiva en computación que implica introducir un conjunto de datos existente (a menudo marcado o etiquetado), iterar sobre el

conjunto de datos y después ajustar los parámetros del modelo, tales como pesos y sesgos, para mejorar el rendimiento del modelo. Generalmente, la optimización basada en una predicción incorrecta implica propagarse de nuevo a través de las capas del modelo y determinar gradientes para reducir el error de predicción.

Las NPU diseñadas para acelerar la inferencia generalmente se configuran para operar en modelos completos. Por lo tanto, tales NPU pueden configurarse para ingresar una nueva pieza de datos y procesar rápidamente esta nueva pieza a través de un modelo ya entrenado para generar una salida de modelo (p. ej., una inferencia).

En algunas implementaciones, la NPU 908 es una parte de una o más de la CPU 902, la GPU 904 y/o el DSP 906. Estos pueden estar ubicados en un equipo de usuario (UE, en inglés) en un sistema de comunicación inalámbrica u otro dispositivo informático.

En algunos ejemplos, el componente de conectividad 912 puede incluir subcomponentes, por ejemplo, para conectividad de tercera generación (3G), conectividad de cuarta generación (4G) (p. ej., LTE), conectividad de quinta generación (5G) (p. ej., NR), conectividad Wi-Fi, conectividad Bluetooth y otros estándares de transmisión inalámbrica de datos. El componente de conectividad 912 puede acoplarse adicionalmente a una o más antenas 914.

El sistema de procesamiento 900 también puede incluir una o más unidades de procesamiento de sensores 916 asociadas con cualquier manera de sensor, uno o más procesadores de señal de imagen (ISP) 918 asociados con cualquier manera de sensor de imagen, y/o un procesador de navegación 920, que puede incluir componentes del sistema de posicionamiento basado en satélite (p. ej., GPS o GLONASS) así como componentes del sistema de posicionamiento inercial.

El sistema de procesamiento 900 también puede incluir uno o más dispositivos de entrada y/o salida 922, tales como pantallas, superficies sensibles al tacto (incluidas pantallas sensibles al tacto), botones físicos, altavoces, micrófonos y lo similar.

En algunos ejemplos, uno o más de los procesadores del sistema de procesamiento 900 pueden basarse en un conjunto de instrucciones ARM o RISC-V.

El sistema de procesamiento 900 también incluye la memoria 924, que es representativa de una o más memorias estáticas y/o dinámicas, tales como una memoria dinámica de acceso aleatorio, una memoria estática basada en flash, y lo similar. En este ejemplo, la memoria 924 incluye componentes ejecutables por computador, que pueden ejecutarse por uno o más de los procesadores antes mencionados del sistema de procesamiento 900.

En particular, en este ejemplo, la memoria 924 incluye un componente generador de conjunto de tokens 924A,

un componente de selección de token 924B, un componente de generación de salida 924C y un modelo de inteligencia artificial generativa 924D. Los componentes representados, y otros no representados, pueden configurarse para realizar
5 diversos aspectos de los métodos divulgados en la presente.

Generalmente, el sistema de procesamiento 900 y/o componentes de este pueden configurarse para realizar los métodos divulgados en la presente.

10 Cláusulas de ejemplo

Los detalles de implementación de diversos aspectos de la presente divulgación se describen en las siguientes cláusulas numeradas.

Cláusula 1: Un método implementado por procesador,
15 que comprende: recibir una pluralidad de conjuntos de tokens generados basado en un indicador de entrada y un primer modelo de inteligencia artificial generativa, cada conjunto de tokens en la pluralidad de conjuntos de tokens correspondiente a una respuesta candidata al indicador de
20 entrada; seleccionar, usando un segundo modelo de inteligencia artificial generativa y ajuste recursivo de una distribución objetivo asociada con la pluralidad recibida de conjuntos de tokens, un conjunto de tokens de la pluralidad de conjuntos de tokens; y emitir el conjunto de tokens
25 seleccionado como una respuesta al indicador de entrada.

Cláusula 2: El método de la cláusula 1, en donde la pluralidad de conjuntos de tokens se organizan en una estructura de datos de árbol.

Cláusula 3: El método de la cláusula 2, en donde
5 el indicador de entrada corresponde a un nodo raíz de la estructura de datos de árbol, y cada ruta a través de la estructura de datos de árbol corresponde a una respuesta candidata diferente al indicador de entrada.

Cláusula 4: El método de la cláusula 2 o 3, en
10 donde un número de tokens a un nivel en la estructura de datos de árbol está basado en un factor de ramificación asociado con un nivel inmediatamente anterior al nivel en la estructura de datos de árbol.

Cláusula 5: El método de cualquiera de las
15 cláusulas 2 a 4, en donde una profundidad de la estructura de datos de árbol corresponde a un número máximo de tokens generados por un único pase a través del primer modelo de inteligencia artificial generativa.

Cláusula 6: El método de cualquiera de las
20 cláusulas 1 a 5, en donde un tamaño de cada conjunto de tokens está basado en una métrica de complejidad computacional asociada con generar un conjunto de tokens objetivo por el segundo modelo de inteligencia artificial generativa.

25 Cláusula 7: El método de cualquiera de las cláusulas 1 a 6, en donde el ajuste recursivo de la

distribución objetivo comprende: determinar si aceptar o rechazar un primer token en un conjunto de tokens de la pluralidad de conjuntos de tokens; y ajustar una distribución de probabilidad usada para verificar un segundo token en el conjunto de tokens posterior al primer token basado en la determinación de si aceptar o rechazar el primer token.

Cláusula 8: El método de la cláusula 7, en donde ajustar la distribución de probabilidad comprende restar un valor de probabilidad asociado con el primer token de la distribución de probabilidad basado en determinar rechazar el primer token.

Cláusula 9: El método de cualquiera de las cláusulas 1 a 8, en donde seleccionar el conjunto de tokens de la pluralidad de conjuntos de tokens comprende: rechazar un primer token en un primer nivel de una estructura de datos de árbol que representa la pluralidad de conjuntos de tokens; emitir una distribución de probabilidad ajustada basada en el rechazo del primer token; descartar, de la estructura de datos de árbol, tokens hijos del primer token en niveles más profundos que el primer nivel de la estructura de datos de árbol; y determinar si aceptar o rechazar un segundo token en el primer nivel de la estructura de datos de árbol basado en la distribución de probabilidad ajustada.

Cláusula 10: El método de cualquiera de las cláusulas 1 a 9, en donde seleccionar el conjunto de tokens de la pluralidad de conjuntos de tokens comprende: rechazar

cada conjunto de tokens generado por el primer modelo de inteligencia artificial generativa; y muestrear, usando el segundo modelo generativo de inteligencia artificial, un token basado en una distribución objetivo que excluye las probabilidades asociadas con cada conjunto de tokens
5 generado por el primer modelo generativo de inteligencia artificial, en donde el conjunto de tokens seleccionado comprende el token muestreado.

Cláusula 11: El método de cualquiera de las
10 cláusulas 1 a 10, en donde: el primer modelo de inteligencia artificial generativa corresponde a un modelo de borrador en una canalización de decodificación especulativa, y el segundo modelo de inteligencia artificial generativa corresponde a un modelo objetivo en
15 la canalización de decodificación especulativa.

Cláusula 12: Un método implementado por procesador, que comprende: generar, basado en un indicador de entrada y un modelo de inteligencia artificial generativa, una primera pluralidad de conjuntos de tokens, cada conjunto
20 de tokens en la primera pluralidad de conjuntos de tokens correspondiente a una primera porción de una respuesta candidata al indicador de entrada; generar especulativamente, usando el modelo de inteligencia artificial generativa, una segunda pluralidad de conjuntos de tokens, cada conjunto de
25 tokens en la segunda pluralidad de conjuntos de tokens correspondiente a una segunda porción de la respuesta

candidata al indicador de entrada basado en la primera pluralidad de conjuntos de tokens; mientras que se genera especulativamente la segunda pluralidad de conjuntos de tokens, seleccionar un conjunto de tokens de la primera pluralidad de conjuntos de tokens; y emitir el conjunto de tokens seleccionado de la primera pluralidad de tokens y un conjunto de tokens asociado en la segunda pluralidad de tokens como una respuesta al indicador de entrada.

Cláusula 13: El método de la cláusula 12, en donde seleccionar el conjunto de tokens de la primera pluralidad de conjuntos de tokens comprende seleccionar una secuencia más larga de tokens aceptados de la primera pluralidad de conjuntos de tokens.

Cláusula 14: El método de la cláusula 12 o 13, en donde los conjuntos de tokens en la segunda pluralidad de conjuntos de tokens incluyen protección responsable de un número de tokens en el conjunto de tokens seleccionado de la primera pluralidad de conjuntos de tokens.

Cláusula 15: El método de cualquiera de las cláusulas 12 a 14, en donde la primera pluralidad de conjuntos de tokens se representan como una estructura de datos de árbol y un nodo raíz de la estructura de datos de árbol corresponde al indicador de entrada.

Cláusula 16: El método de la cláusula 15, en donde una profundidad de la estructura de datos de árbol corresponde a un número máximo de tokens generados por un

único pase a través del modelo de inteligencia artificial generativa.

Cláusula 17: El método de la cláusula 15 o 16, en donde se establece un tamaño máximo de la estructura de
5 datos de árbol basado en una métrica de complejidad computacional asociada con generar un conjunto de tokens por el modelo de inteligencia artificial generativa.

Cláusula 18: El método de cualquiera de las cláusulas 15 a 17, en donde seleccionar el conjunto de tokens
10 de la primera pluralidad de conjuntos de tokens comprende: rechazar un primer token en un primer nivel de la estructura de datos de árbol que representa la primera pluralidad de conjuntos de tokens; emitir una distribución de probabilidad ajustada basada en el rechazo del primer token; descartar
15 tokens hijos del primer token de la estructura de datos de árbol; y determinar si aceptar o rechazar un segundo token en el primer nivel de la estructura de datos de árbol basado en la distribución de probabilidad ajustada.

Cláusula 19: El método de cualquiera de las
20 cláusulas 12 a 18, en donde seleccionar el conjunto de tokens de la primera pluralidad de conjuntos de tokens comprende: rechazar cada conjunto de tokens en la primera pluralidad de conjuntos de tokens generados por el modelo de inteligencia artificial generativa; y muestrear, usando
25 el modelo de inteligencia artificial generativa, un token basado en una distribución objetivo que excluye las

probabilidades asociadas con cada conjunto de tokens en la primera pluralidad de conjuntos de tokens, en donde el conjunto de tokens seleccionado de la primera pluralidad de conjuntos de tokens comprende el token muestreado.

5 Cláusula 20: El método de cualquiera de las cláusulas 12 a 19, en donde el modelo de inteligencia artificial generativa comprende un modelo de inteligencia artificial generativa entrenado para generar múltiples tokens en respuesta al indicador de entrada basado en las
10 incrustaciones de indicadores de pronósticos.

 Cláusula 21: El método de cualquiera de las cláusulas 12 a 20, en donde el modelo de inteligencia artificial generativa comprende un modelo que incluye una o más capas no autorregresivas y una o más capas
15 autorregresivas.

 Cláusula 22: El método de la cláusula 21, en donde la una o más capas autorregresivas comprenden una o más capas en una parte superior de una pila de capas que representan el modelo de inteligencia artificial generativa.

20 Cláusula 23: El método de la cláusula 21 o 22, en donde la una o más capas autorregresivas comprenden una o más capas en la parte inferior de una pila de capas que representan el modelo de inteligencia artificial generativa.

 Cláusula 24: Un sistema de procesamiento, que
25 comprende: al menos una memoria que tiene instrucciones ejecutables almacenadas en esta; y uno o más procesadores

configurados para ejecutar las instrucciones ejecutables para hacer que el sistema de procesamiento realice las operaciones de cualquiera de las cláusulas 1 a 23.

Cláusula 25: Un sistema de procesamiento, que
5 comprende: medios para realizar las operaciones de cualquiera de las cláusulas 1 a 23.

Cláusula 26: Un medio legible por computador no transitorio que tiene instrucciones ejecutables almacenadas en este que, cuando se ejecuta por uno o más
10 procesadores, realiza las operaciones de cualquiera de las cláusulas 1 a 23.

Consideraciones adicionales

La descripción anterior se proporciona para
15 permitir que cualquier experto en la técnica ponga en práctica los diversos aspectos divulgados en la presente. Los ejemplos analizados en la presente no son limitantes del alcance, aplicabilidad o aspectos establecidos en las reivindicaciones. Diversas modificaciones a estos aspectos
20 serán fácilmente evidentes para los expertos en la técnica, y los principios genéricos definidos en la presente pueden aplicarse a otros aspectos. Por ejemplo, se pueden realizar cambios en la función y disposición de los elementos analizados sin apartarse del alcance de la divulgación.
25 Varios ejemplos pueden omitir, sustituir o añadir diversos procedimientos o componentes según sea apropiado. Por

ejemplo, los métodos divulgados pueden realizarse en un orden diferente al descrito, y pueden añadirse, omitirse o combinarse diversas etapas. Además, las características descritas con respecto a algunos ejemplos pueden combinarse en algunos otros ejemplos. Por ejemplo, se puede implementar un aparato o se puede poner en práctica un método usando cualquier número de los aspectos expuestos en la presente. Además, el alcance de la divulgación está destinado a cubrir dicho aparato o método que se practica usando otra estructura, funcionalidad o estructura y funcionalidad además de, o que no sean, los diversos aspectos de la divulgación establecidos en la presente. Debe entenderse que cualquier aspecto de la divulgación que se describe en la presente puede ser realizado mediante uno o más elementos de una reivindicación.

Como se usa en la presente, la palabra “ejemplar” significa “servir como ejemplo, instancia o ilustración”. Cualquier aspecto divulgado en la presente como “ilustrativo” no debe interpretarse necesariamente como preferido o ventajoso sobre otros aspectos.

Como se usa en la presente, una frase que se refiere a “al menos uno de” una lista de elementos se refiere a cualquier combinación de esos elementos, incluidos los miembros individuales. Como ejemplo, “al menos uno de: a, b o c” pretende cubrir a, b, c, a-b, a-c,

b-c y a-b-c, así como cualquier combinación con múltiplos del mismo elemento (p. ej., a-a, a-a-a, a-a-b, a-a-c, a-b-b, a-c-c, b-b, b-b-b, b-b-c, c-c, y c-c-c o cualquier otro ordenamiento de a, b, y c).

5 Como se usa en la presente, el término “determinar” abarca una amplia variedad de acciones. Por ejemplo, “determinar” puede incluir calcular, computar, procesar, derivar, investigar, buscar (p. ej., buscar en una tabla, una base de datos u otra estructura de datos),
10 determinar, y lo similar. Además, “determinar” puede incluir recibir (p. ej., recibir información), acceder (p. ej., acceder a datos en una memoria), y lo similar. También, “determinar” puede incluir resolver, seleccionar, elegir, establecer, y lo similar.

15 Los métodos divulgados en la presente comprenden una o más etapas o acciones para lograr los métodos. Las etapas o acciones del método pueden intercambiarse entre sí sin apartarse del alcance de las reivindicaciones. Es decir, a menos que se especifique un orden específico de etapas o
20 acciones, el orden y/o uso de etapas y/o acciones específicas podrá modificarse sin apartarse del alcance de las reivindicaciones. Además, las diversas operaciones de los métodos descritos anteriormente pueden ser realizadas por cualquier medio adecuado capaz de realizar las funciones
25 correspondientes. Los medios pueden incluir diversos componente(s) y/o módulo(s) de hardware y/o software, que

incluyen, pero no se limitan a, un circuito, un circuito integrado específico de aplicación (ASIC, en inglés) o un procesador. Generalmente, donde hay operaciones ilustradas en figuras, esas operaciones pueden tener componentes de
5 medias más función de contraparte correspondientes con numeración similar.

Las siguientes reivindicaciones no pretenden limitarse a los aspectos mostrados en la presente, sino que deben concederse el alcance completo consistente con el
10 lenguaje de las reivindicaciones. Dentro de una reivindicación, la referencia a un elemento en singular no pretende significar “uno y solo uno” a menos que se indique específicamente de esta manera, sino en su lugar “uno o más”. A menos que se indique específicamente de cualquier otra
15 manera, el término “algunos” se refiere a uno o más. Ningún elemento de reclamación debe interpretarse en virtud de lo dispuesto en 35 U.S.C. § 112 (f) a menos que el elemento se recite expresamente usando la frase “medios para” o, en el caso de una reclamación de método, el elemento se mencione
20 usando la frase “paso para”. Todos los equivalentes estructurales y funcionales de los elementos de los diversos aspectos descritos a lo largo de esta divulgación que se conocen o que más tarde llegan a conocerse por los expertos en la técnica se incorporan expresamente en la presente por
25 referencia y se pretende que estén abarcada por las

reivindicaciones. Además, nada de lo que se divulga en la presente está destinado a ser dedicado al público, independientemente de si dicha divulgación se menciona explícitamente en las reivindicaciones.