(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 372 863**

(51) Int. CI.:
G10L 15/18 (2006.01)
G06F 17/22 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Número de solicitud europea: **06772381 .7**
(96) Fecha de presentación: **06.06.2006**
(97) Número de publicación de la solicitud: **1891545**
(97) Fecha de publicación de la solicitud: **27.02.2008**

(54) Título: **COMPRESIÓN DE MODELOS DE LENGUAJE CON CODIFICACIÓN GOLOMB.**

(30) Prioridad:
23.06.2005 US 159712

(45) Fecha de publicación de la mención BOPI:
27.01.2012

(45) Fecha de la publicación del folleto de la patente:
27.01.2012

(73) Titular/es:
MICROSOFT CORPORATION
ONE MICROSOFT WAY
REDMOND, WASHINGTON 98052-6399, US

(72) Inventor/es:
CHURCH, Kenneth Ward;
THIESSON, Bo y
HART, Edward C. Jr.

(74) Agente: **Carpintero López, Mario**

ES 2 372 863 T3

DESCRIPCIÓN

Compresión de modelos de lenguaje con codificación Golomb

La explicación dada a continuación se proporciona meramente como información de antecedentes general y no se pretende que se use como una ayuda al determinar el alcance de la materia objeto que se reivindica.

Los modelos de lenguaje se usan en una variedad de aplicaciones incluyendo aplicaciones de canal con ruido tales como procesamiento de lenguaje natural, revisión ortográfica y similares. En las aplicaciones de lenguaje natural, normalmente actúa un reconocedor de voz que combina evidencia acústica (modelo de canal) con unas expectativas acerca de lo que es probable que diga el usuario (modelo de lenguaje). Se hace referencia a una forma común de modelos de lenguaje como un tri–grama.

En general, un n–grama es una secuencia secundaria de n testigos (palabras). Un tri–grama es una secuencia secundaria de 3 testigos. Por ejemplo, a partir de la expresión “to be or not to be”, pueden generarse 8 tri–gramas: “$ $ to”, “$ to be”, “to be or” “be or not”, “or not to”, “not to be”, “to be $” y “be $ $”, en los que la cadena de entrada se rellena con dos testigos especiales que se indican en: “$”. La estadística puede aplicarse a tales n–gramas para estimar una probabilidad de que un usuario pretendiera una entrada particular.

A pesar de que mil millones de palabras de texto solía considerarse una cantidad grande, los conjuntos de entrenamiento para reconocimiento de voz se entrenan de forma rutinaria con diez mil millones de palabras de texto. En general, los modelos de lenguaje grandes funcionan bien (lo que quiere decir que tienen una baja entropía); no obstante, la capacidad de memoria es a menudo limitada, especialmente en dispositivos móviles tales como teléfonos móviles, asistentes digitales personales (PDA), agendas electrónicas y similares. Una técnica para ocuparse de la situación de la memoria implica el recorte del modelo de lenguaje, eliminando palabras que se usan con poca frecuencia y variantes poco comunes. No obstante, la eliminación de tales términos reduce la efectividad global del modelo de lenguaje, lo que conduce a más errores de semántica debido a la incapacidad de hacer que se corresponda la entrada con las palabras en el modelo recortado.

La patente US 6.092.038 A se refiere a técnicas para proporcionar una compresión sin pérdida de modelos de lenguaje n–gramas en un decodificador en tiempo real. En un ejemplo, una serie de palabras de registros de n–grama se asigna a números de palabra. Una diferencia entre números de palabra posteriores de los registros de n–grama se calcula a continuación y estas diferencias se almacenan de tal modo que las diferencias ocupan unos bloques más pequeños de memoria que los números de palabra.

El documento WO 01/10036 A1 se refiere a un sistema de codificación/ decodificación de longitud variable eficiente en cuanto a la memoria. Para evitar el almacenamiento de grandes tablas de consulta, se han desarrollado las técnicas de codificación Golomb. Se puede pensar en los códigos Golomb como en un conjunto especial de palabras de código libres de prefijo de longitud variable que se optimiza para unos números no negativos que tienen una distribución de probabilidad geométrica que disminuye de forma exponencial. Las palabras de código se construyen de tal modo que pueden decodificarse directamente sin la necesidad de una tabla de consulta.

El documento US 2004/138884 se refiere a modelos de compresión de lenguaje y a identificadores de palabras que se usan por tales sistemas.

El documento US 6.169.969 B1 se refiere al campo del establecimiento de correspondencia de cadena rápido y masivo.

Sumario

El objeto de la presente invención es la mejora de los sistemas de la técnica anterior. Este objeto se soluciona mediante la materia objeto de las reivindicaciones independientes. Las realizaciones preferidas se definen mediante las reivindicaciones dependientes.

Este sumario se proporciona para introducir de una forma simplificada algunos conceptos, que se describen a continuación en la Descripción detallada. No se pretende que este Sumario identifique los rasgos clave o las características esenciales de la materia objeto que se reivindica, ni se pretende que se use como una ayuda al determinar el alcance de la materia objeto que se reivindica.

En una realización, un modelo de lenguaje se comprime usando técnicas de codificación Golomb. Una lista de valores se genera a partir de elementos del modelo de lenguaje. La lista de valores enteros se ordena, y para cada elemento, se calcula una diferencia entre valores enteros adyacentes en la lista. Cada diferencia calculada se codifica usando un código Golomb.

En otra realización, un sistema para procesar entradas de usuario tiene una interfaz de usuario, una memoria, un codificador/ decodificador Golomb, y un procesador. La interfaz de usuario está adaptada para recibir entradas de usuario. La memoria está adaptada para almacenar información y para almacenar un modelo de lenguaje comprimido por codificación Golomb. El codificador/ decodificador Golomb está adaptado para codificar una entrada

de usuario y para decodificar elementos del modelo de lenguaje comprimido por codificación Golomb. El procesador está adaptado para comparar una entrada de usuario codificada frente a elementos del modelo de lenguaje comprimido por codificación Golomb para identificar las correspondencias probables.

En otra realización, se proporciona un procedimiento de decodificación de entradas de usuario usando un modelo de lenguaje codificado por codificación Golomb. Una entrada de usuario se divide en una pluralidad de elementos, cada uno de los cuales se codifica usando una técnica de función hash. Cada elemento codificado se compara con unos elementos de un modelo de lenguaje codificado por codificación Golomb para identificar las correspondencias posibles. Las correspondencias posibles se analizan estadísticamente para estimar una probabilidad de que una correspondencia posible sea una asignación correcta de la entrada de usuario para el modelo de lenguaje codificado por codificación Golomb.

Breve descripción de los dibujos

La figura 1 es un diagrama de bloques de un entorno de cálculo en el que pueden ponerse en práctica las realizaciones.
La figura 2 es un diagrama de bloques de un entorno de cálculo alternativo en el que pueden ponerse en práctica las realizaciones.
La figura 3 es un diagrama de flujo simplificado de una realización de un proceso para la compresión de un modelo de lenguaje para su uso en dispositivos de cálculo.
La figura 4 es un diagrama de flujo simplificado de un proceso para la codificación Golomb de diferencias entre unos valores de función hash que se calculan de acuerdo con el proceso de la figura 3.
La figura 5 es un diagrama de bloques simplificado de un árbol de Huffman que ilustra un código unario.
La figura 6 es un diagrama de flujo simplificado de una realización de un proceso para la decodificación de una primera diferencia codificada por codificación Golomb.
La figura 7 es un diagrama de bloques simplificado de una realización de un sistema adaptado para usar un modelo de lenguaje comprimido con técnicas de codificación Golomb.
La figura 8 es un diagrama de flujo simplificado de una realización de un proceso para la decodificación de entrada de usuario frente a un modelo de lenguaje codificado por codificación Golomb.

Descripción detallada

Los modelos de lenguaje se utilizan en sistemas de reconocimiento de voz, en sistemas de corrección ortográfica sensibles al contexto, en interfaces que se usan para introducir caracteres asiáticos en los ordenadores y similares. Las técnicas de compresión Golomb pueden aplicarse a entradas de usuario, tales como los datos de localizador de recursos uniforme (URL) para la navegación en redes informáticas globales, tal como Internet. Debido a que la memoria está a menudo limitada en la práctica, especialmente en plataformas móviles tales como teléfonos móviles, asistentes digitales personales (PDA) y similares, la compresión del modelo de lenguaje puede ser bastante útil, y las técnicas de codificación Golomb pueden usarse tanto para comprimir un modelo de lenguaje como para decodificar los resultados.

La figura 1 ilustra un ejemplo de un entorno 100 de sistema de cálculo adecuado sobre el que pueden llevarse a cabo las técnicas de compresión de modelo de lenguaje de las realizaciones. El entorno 100 de sistema de cálculo es sólo un ejemplo de un entorno de cálculo adecuado y no se pretende sugerir ninguna limitación en lo que concierne al alcance del uso o la funcionalidad de la invención. Tampoco ha de interpretarse el entorno 100 de cálculo tal como si tuviera ninguna dependencia o requisito en relación con ninguno de los componentes o ninguna combinación de los mismos que se ilustra en el entorno 100 operativo a modo de ejemplo.

Las realizaciones de la invención son funcionales con numerosas otros entornos o configuraciones de sistema de cálculo de general propósito o de propósito especial. Ejemplos de entornos, configuraciones y/o sistemas de cálculo bien conocidos que pueden ser adecuados para su uso con las realizaciones de la invención incluyen, pero no se limitan a, ordenadores personales, equipos servidores, dispositivos de mano o portátiles, sistemas multiprocesador, sistemas basados en microprocesador, decodificadores, electrónica de consumo programable, PC en red, miniordenadores, ordenadores de gran sistema, sistemas de telefonía, entornos de cálculo distribuido que incluyen cualquiera de los sistemas o dispositivos anteriores y similares.

Las realizaciones pueden describirse en el contexto general de unas instrucciones ejecutables por ordenador, tal como módulos de programa, que se ejecutan por un ordenador. En general, los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos, etc. que realizan unas tareas particulares o implementan unos tipos de datos abstractos particulares. La invención se diseña para ponerse en práctica en unos entornos de cálculo distribuidos en los que las tareas se realizan mediante unos dispositivos de procesamiento remoto que están enlazados a través de una red de comunicaciones. En un entorno de cálculo distribuido, los módulos de programa se encuentran en unos medios de almacenamiento informático tanto locales como remotos lo que incluye los dispositivos de almacenamiento de memoria.

Con referencia a la figura 1, un sistema a modo de ejemplo para implementar una realización incluye un dispositivo de cálculo de propósito general en la forma de un ordenador 110. Los componentes del ordenador 110 pueden

incluir, pero no se limitan a, una unidad 120 de procesamiento, una memoria 130 de sistema, y un bus 121 de sistema que acopla varios componentes de sistema lo que incluye la memoria de sistema a la unidad 120 de procesamiento. El bus 121 de sistema puede ser cualquiera de diversos tipos de estructuras de bus lo que incluye un bus de memoria o un controlador de memoria, un bus de periféricos, y un bus local que usa cualquiera de una variedad de arquitecturas de bus. A modo de ejemplo, y no de limitación, tales arquitecturas incluyen el bus *Industry Standard Architecture* (ISA, Arquitectura Estándar de la Industria), el bus *Micro Channel Architecture* (MCA, Arquitectura de Microcanal), el bus *Enhanced ISA* (EISA, ISA mejorado), el bus local *Video Electronics Standards Association* (VESA, Asociación de Normalización de Electrónica y Vídeo) y el bus *Peripheral Component Interconnect* (PCI, Interconexión de Componentes Periféricos) que también se conoce como bus Mezzanine.

El ordenador 110 normalmente incluye una variedad de medios legibles por ordenador. Los medios legibles por ordenador pueden ser cualesquiera medios disponibles a los que puede accederse mediante el ordenador 110 y que incluyen unos medios tanto volátiles como no volátiles, medios extraíbles y no extraíbles. A modo de ejemplo, y no de limitación, los medios legibles por ordenador pueden comprender unos medios de almacenamiento informático y unos medios de comunicación. Los medios de almacenamiento informático incluyen unos medios tanto volátiles como no volátiles, tanto extraíbles como no extraíbles que se implementan en cualquier procedimiento o tecnología para el almacenamiento de información tal como instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos. Los medios de almacenamiento informático incluyen, pero no se limitan a, RAM, ROM, EEPROM, memoria flash u otra tecnología de memoria, CD–ROM, discos versátiles digitales (DVD) u otro almacenamiento en disco óptico, casetes magnéticos, cinta magnética, almacenamiento en disco magnético u otros dispositivos de almacenamiento magnético, o cualquier otro medio que pueda usarse para almacenar la información deseada y al que pueda accederse mediante el ordenador 110. Los medios de comunicación normalmente incorporan instrucciones legibles por ordenador, estructuras de datos, módulos de programa u otros datos en una señal de datos modulada tal como una onda portadora u otro mecanismo de transporte e incluyen cualesquiera medios de entrega de información. La expresión “señal de datos modulada” significa una señal que tiene una o más de sus características establecidas o cambiadas de una forma tal que se codifica una información en la señal. A modo de ejemplo, y no de limitación, los medios de comunicación incluyen unos medios por cable tales como una red por cable o una conexión por cable directa, y medios inalámbricos tales como acústicos, de RF, por infrarrojos y otros medios inalámbricos. Han de incluirse también las combinaciones de cualquiera de los anteriores dentro del alcance de los medios legibles por ordenador.

La memoria 130 de sistema incluye unos medios de almacenamiento informático en la forma de memoria volátil y/o no volátil tal como una memoria de sólo lectura (ROM) 131 y una memoria de acceso aleatorio (RAM) 132. Un sistema 133 básico de entrada/salida (BIOS), que contiene las rutinas básicas que ayudan a la transferencia de la información entre los elementos dentro del ordenador 110, tal como durante el arranque, se almacena normalmente en la ROM 131. La RAM 132 normalmente contiene unos datos y/o módulos de programa a los que puede accederse inmediatamente y/o sobre los que opera en dicho momento la unidad 120 de procesamiento. A modo de ejemplo, y no de limitación, la figura 1 ilustra un sistema 134 operativo, unos programas 135 de aplicación, otros módulos 136 de programa, y los datos 137 de programa.

El ordenador 110 puede incluir también otros medios de almacenamiento informático extraíbles/ no extraíbles volátiles/ no volátiles. Sólo a modo de ejemplo, la figura 1 ilustra una unidad 141 de disco duro que lee a partir de o que escribe en unos medios magnéticos no extraíbles, no volátiles, una unidad 151 de disco magnético que lee a partir de o que escribe en un disco 152 magnético extraíble, no volátil, y una unidad 155 de disco óptico que lee a partir de o que escribe en un disco 156 óptico extraíble no volátil tal como un CD ROM u otros medios ópticos. Otros medios de almacenamiento informático extraíbles/ no extraíbles, volátiles/ no volátiles que pueden usarse en el entorno operativo a modo de ejemplo incluyen, pero no se limitan a, casetes de cinta magnética, tarjetas de memoria flash, discos versátiles digitales, cinta de vídeo digital, RAM de estado sólido, ROM de estado sólido y similares. La unidad 141 de disco duro se conecta normalmente al bus 121 de sistema a través de una interfaz de memoria no extraíble tal como la interfaz 140, y una unidad 151 de disco magnético y una unidad 155 de disco óptico se conectan normalmente al bus 121 de sistema mediante una interfaz de memoria extraíble, tal como la interfaz 150.

Las unidades y sus medios asociados de almacenamiento informático que se discuten anteriormente y que se ilustran en la figura 1, proporcionan un almacenamiento de instrucciones legibles por ordenador, estructuras de datos, módulos de programa y otros datos para el ordenador 110. En la figura 1, por ejemplo, la unidad 141 de disco duro se ilustra como un sistema 144 operativo de almacenamiento, unos programas 145 de aplicación, otros módulos 146 de programa, y unos datos de 147 programa. Obsérvese que estos componentes pueden ser o bien los mismos que o bien diferentes del sistema 134 operativo, los programas 135 de aplicación, otros módulos 136 de programa, y los datos 137 de programa. En este caso, se dan al sistema operativo 144, a los programas 145 de aplicación, a los otros módulos 146 de programa, y a los datos de 147 programa unos números diferentes para ilustrar que, como mínimo, son copias diferentes.

Un usuario puede introducir órdenes e información en el ordenador 110 a través de unos dispositivos de entrada tales como un teclado 162, un micrófono 163, y un dispositivo 161 señalador, tal como un ratón, una bola de seguimiento o un teclado táctil. Otros dispositivos de entrada (que no se muestran) pueden incluir un joystick, un controlador para juegos, una antena parabólica, un escáner, o similar. Estos y otros dispositivos de entrada se conectan a menudo a la unidad 120 de procesamiento a través de una interfaz 160 de entrada de usuario que se

acopla al bus de sistema, pero puede estar conectada mediante otra interfaz y estructuras de bus, tal como un puerto paralelo, un puerto para juegos o un bus serie universal (USB). Un monitor 191 u otro tipo de dispositivo de visualización se conecta también al bus 121 de sistema a través de una interfaz, tal como una interfaz 190 de vídeo. Además del monitor, los ordenadores pueden incluir también otros dispositivos de salida de periféricos tales como unos altavoces 197 y una impresora 196, que puede estar conectada a través de una interfaz 195 de periféricos de salida.

El ordenador 110 se hace funcionar en un entorno de red que usa unas conexiones lógicas a uno o más ordenadores remotos, tal como un ordenador 180 remoto. El ordenador 180 remoto puede ser un ordenador personal, un dispositivo de mano, un servidor, un enrutador, un PC de red, un dispositivo del mismo nivel u otro nodo de red común, y normalmente incluye muchos o la totalidad de los elementos que se describen anteriormente en relación con el ordenador 110. Las conexiones lógicas que se muestran en la figura 1 incluyen una red de área local (LAN) 171 y una red de área amplia (WAN) 173, pero pueden incluir también otras redes. Tales entornos de red son muy frecuentes en oficinas, redes de ordenadores por toda la empresa, intranet e Internet.

Cuando se usa en un entorno de red LAN, el ordenador 110 se conecta a la LAN 171 a través de un adaptador o interfaz de red 170. Cuando se usa en un entorno de red WAN, el ordenador 110 normalmente incluye un módem 172 u otros medios para el establecimiento de comunicaciones a lo largo de la WAN 173, tal como Internet. El módem 172, que puede ser interno o externo, puede estar conectado al bus 121 de sistema a través de la interfaz 160 de entrada de usuario, o de otro mecanismo adecuado. En un entorno de red, los módulos de programa que se muestran en relación con el ordenador 110, o con partes del mismo, pueden almacenarse en el dispositivo de almacenamiento de memoria remoto. A modo de ejemplo, y no de limitación, la figura 1 ilustra unos programas 185 de aplicación remota como que residen en el ordenador 180 remoto. Se apreciará que las conexiones de red que se muestran son a modo de ejemplo y que pueden usarse otros medios de establecimiento de un enlace de comunicaciones entre los ordenadores.

La figura 2 es un diagrama de bloques de un dispositivo 200 móvil, que es un entorno de cálculo a modo de ejemplo. El dispositivo 200 móvil incluye un microprocesador 202, una memoria 204, unos componentes 206 de entrada/salida (E/S), y una interfaz 208 de comunicación para la comunicación con ordenadores remotos o con otros dispositivos móviles. En una realización, los componentes que se mencionan anteriormente se acoplan para la comunicación entre sí a lo largo de un bus 210 adecuado.

La memoria 204 se implementa como una memoria electrónica no volátil tal como una memoria de acceso aleatorio (RAM) con un módulo de reserva de batería (que no se muestra) de tal modo que la información que se almacena en la memoria 204 no se pierde cuando se apaga la alimentación general al dispositivo 200 móvil. Una parte de la memoria 204 se asigna preferiblemente como una memoria direccionable para la ejecución de programas, mientras que otra parte de la memoria 204 se usa preferiblemente para el almacenamiento, tal como para simular el almacenamiento en una unidad de disco.

La memoria 204 incluye un sistema 212 operativo, unos programas 214 de aplicación así como un almacenamiento 216 de objetos. Durante el funcionamiento, el sistema 212 operativo se ejecuta preferiblemente mediante el procesador 202 a partir de la memoria 204. El sistema 212 operativo, en una realización preferida, es un sistema operativo de marca WINDOWS® CE comercialmente disponible de Microsoft Corporation. El sistema 212 operativo se diseña preferiblemente para dispositivos móviles, e implementa unas características de base de datos que pueden utilizarse por las aplicaciones 214 a través de un conjunto de interfaces y de métodos de programación de aplicaciones expuestas. Los objetos en el almacenamiento 216 de objetos se mantienen por las aplicaciones 214 y el sistema 212 operativo, al menos en parte en respuesta a unas llamadas a las interfaces y a los métodos de programación de aplicaciones expuestas.

La interfaz 208 de comunicación representa numerosos dispositivos y tecnologías que permiten que el dispositivo 200 móvil envíe y reciba información. Los dispositivos incluyen módems por cable e inalámbricos, receptores por satélite y sintonizadores de difusión, por nombrar unos pocos. El dispositivo 200 móvil puede conectarse también directamente a un ordenador para intercambiar datos entre los mismos. En tales casos, la interfaz 208 de comunicación puede ser un transceptor de infrarrojos o una conexión de comunicación en serie o en paralelo, la totalidad de los cuales son capaces de transmitir una información de transmisión por secuencias.

Los componentes 206 de entrada/salida incluyen una variedad de dispositivos de entrada tal como una pantalla sensible al tacto, botones, rodillos, y un micrófono así como una variedad de dispositivos de salida incluyendo un generador de audio, un dispositivo de vibración, y un visualizador. Los dispositivos que se enumeran anteriormente son a modo de ejemplo y no se necesita que todos estén presentes en el dispositivo 200 móvil. Además, otros dispositivos de entrada/salida pueden estar acoplados a, o encontrarse con, el dispositivo 200 móvil dentro del alcance de la presente invención.

Mientras que la reducción del modelo de lenguaje es una técnica posible para ocuparse de las limitaciones de memoria, la compresión es una alternativa más atractiva, debido a que ésta permite el almacenamiento de más que el modelo de lenguaje original en la memoria.

La figura 3 es un diagrama de flujo simplificado de una realización de un proceso para la compresión de un modelo de lenguaje para su uso en dispositivos de cálculo. Un procesador obtiene unos valores numéricos mediante la aplicación de una función hash a unos elementos (tales como palabras, URL, n–gramas y similares), desde 1 hasta P, de acuerdo con la siguiente ecuación

$$H_G = \text{FUNCIÓN HASH (elemento)} \%P,$$

en la que P es un primo adecuado. En una realización, por ejemplo, los valores numéricos pueden ser valores enteros. El término entero tal como se usa en el presente documento se refiere a los números enteros, lo que incluye todos los números naturales, los negativos de estos números, y el cero. Para todos los elementos en el modelo de lenguaje, el procesador obtiene un entero mediante la aplicación de una función hash a cada elemento de una entrada (etapa 300). Los valores de función hash (H_G) se ordenan (etapa 302). Las primeras diferencias (X_G) se calculan entre los resultados de función hash adyacentes (H_G) en la lista (etapa 304). Los intervalos de arribo se almacenan usando un código Golomb (etapa 306). Tal como se usa en el presente documento, la expresión “intervalos de arribo” se refiere a unos espacios o ceros en los datos codificados por codificación Golomb. Por ejemplo, los intervalos de arribo de un proceso de Poisson tienen una distribución exponencial.

Usando este proceso, el número de bits de memoria que se necesitan para almacenar N elementos puede calcularse de acuerdo con la siguiente ecuación:

$$Mbits = N\left[\frac{1}{\log(2)} + \log_2\left(\frac{P}{N}\right)\right],$$

en la que N representa el número de elementos. El primo P representa un equilibrio entre la pérdida y el uso de memoria, tal como el mínimo primo mayor que el resultado de la multiplicación de N por la diferencia deseada promedio entre valores sucesivos. El procedimiento es independiente del tamaño del elemento, de tal modo que las palabras, URL, o n–gramas largos no son más costosos que los cortos (en términos de uso de memoria).

La figura 4 es un diagrama de flujo simplificado de un proceso para la codificación Golomb de diferencias entre unos valores de función hash que se calculan de acuerdo con el proceso de la figura 3. Ha de entenderse que el diagrama de flujo representa un ejemplo de un esquema de codificación Golomb eficiente. Los expertos en la técnica serán capaces de definir otras formas de calcular un cociente y un resto, que tendrán unas características ligeramente diferentes para los requisitos de memoria de la codificación. Un valor K se elige de acuerdo con la siguiente ecuación:

$$k = techo\left(\log_2\left(\frac{1}{2}\times\frac{P}{N}\right)\right),$$

en la que N representa un número de artículos (tal como elementos o n–gramas) en una secuencia de entrada, P representa un primo adecuado más grande que N, tal como el mínimo primo mayor que el resultado de la multiplicación de N por la diferencia deseada promedio entre valores sucesivos, y la función techo indica el entero más pequeño que no es menor que su argumento (etapa 400). Un valor M se elige de acuerdo con la siguiente ecuación:

$$M = 2^K$$

(etapa 402). Un valor de cociente X_q se calcula para cada valor de primera diferencia (X_G) de acuerdo con la siguiente ecuación:

$$X_q = Piso\left(\frac{X_G}{M}\right),\ ),$$

en la que la función piso indica el entero más grande no mayor que su argumento (etapa 404). Un valor de resto X_r se calcula de acuerdo con la siguiente ecuación:

$$X_r = X_G \bmod M$$

para cada uno de los valores de primera diferencia X_G (etapa 406).
El valor de cociente X_q se codifica en formato unario (X_q bits puestos a cero seguidos por un bit puesto a uno) (etapa 408). El resto X_r puede estar codificado en formato binario en K bits (etapa 410). La técnica de codificación Golomb que se ilustra en la figura 4 reduce los requisitos de memoria para almacenar un modelo de lenguaje, de tal modo que el valor de cociente requiere $X_q + 1$ bits, mientras que el valor de resto requiere $\log_2 M$ bits.

Puede considerarse que los valores de función hash ordenados, como una buena aproximación, se han creado mediante un proceso de Poisson. En un proceso de Poisson, los intervalos de arribo tienen una distribución

exponencial. En general, la probabilidad puede expresarse tal como sigue Pr (x) = $\lambda e^{-\lambda x}$, en la que $\lambda = N/P$ y en la que la variable λ representa los intervalos de arribo. El uso de memoria puede calcularse a continuación tal como sigue:

$$H = -\int_0^{\infty} \Pr(x) \log_2 \Pr(x),$$

y el uso de memoria se define mediante la siguiente ecuación:

$$H = \frac{1}{\log_e(2)} + \log_2 \frac{1}{\lambda}.$$

Por lo tanto, el uso de memoria es independiente del tamaño de los elementos. Esto indica que la función hash y la codificación Golomb reducen conjuntamente el uso global de memoria, por ejemplo, de un modelo de lenguaje.

La figura 5 es un diagrama de bloques simplificado de un árbol de Huffman que ilustra un código unario. En este caso, el símbolo A tiene una probabilidad de 0,5, mientras que el símbolo B tiene una probabilidad de 0,25, y así sucesivamente. Supóngase que la probabilidad de x es:

$$Pr\ (x) = (1 - B)\ B^{x}$$

con B = 1/2. El grafo 500 incluye una pluralidad de nodos, con aproximadamente igual probabilidad asignada a cada nodo secundario de un nodo. El nodo 502 raíz representa la palabra o símbolo raíz. El símbolo es o bien una “A” o bien alguna otra cosa. Si el símbolo es una A, se asigna un valor de “1”, lo que se corresponde con la ruta desde la raíz 502 hasta el nodo 506A. El sistema a continuación busca en el siguiente bit o secuencia de bits. Si el símbolo no es una A, el sistema asigna un cero y a continuación comprueba si el símbolo es una B. Si el símbolo es una B, a continuación el sistema asigna un 1 lo que se corresponde con la ruta desde 504B hasta 506B. El valor resultante para un símbolo B es por lo tanto “01”, mientras que el valor resultante para un símbolo A es “1”. En general, el grafo 500 ilustra una secuencia unaria de z – 1 ceros seguidos por un 1.

La tabla 1 a continuación ilustra una probabilidad de un símbolo en base a su posición dentro del grafo en relación con la palabra raíz.

Tabla 1.

Símbolo	Código	Longitud	Pr
A	1	1	2^{-1}
B	01	2	2^{-2}
C	001	3	2^{-2}
N	N (ceros) + 1	N	2^{-N}

La figura 6 es un diagrama de flujo simplificado de un proceso para la decodificación de una primera diferencia codificada por codificación Golomb. Los símbolos del cociente X_q se leen un bit cada vez hasta que se detecta un valor de “1” (etapa 600). Los símbolos del resto X_r se leen en binario (leer $\log_2$ M bits) (etapa 602). Por último, la primera diferencia X se calcula a partir de X_q y X_r tal como sigue:

$$X_G = M^*X_q + X_r$$

que codifica la salida X_G (etapa 604). En este caso, la variable M representa una potencia de dos aproximadamente igual al valor esperado de la primera diferencia dividida por dos y redondeada hacia arriba al entero más cercano en la dirección del infinito positivo.

La figura 7 es un diagrama de bloques simplificado de una realización de un sistema 700 adaptado para la utilización de modelos de lenguaje codificados por codificación Golomb. El sistema 700 incluye una aplicación 702 de software con una interfaz 704 de usuario, un modelo 706 de lenguaje comprimido por codificación Golomb dentro de una memoria 708, un procesador 710, un codificador/ decodificador 712 Golomb, y un conjunto de algoritmos 714 estadísticos. Un usuario introduce unos datos en el sistema 700 a través de la interfaz 704 de usuario. Un codificador/ decodificador 712 Golomb codifica la entrada de usuario y pasa la entrada codificada al procesador 710, que analiza la entrada codificada frente al modelo 706 de lenguaje comprimido de Golomb para producir un conjunto de correspondencias posibles. El procesador 710 usa los algoritmos 714 estadísticos para seleccionar una o más correspondencias probables en base a las palabras dentro del modelo 706 de lenguaje comprimido y pasa las una o más correspondencias probables a la interfaz 704 de usuario como salidas para su visualización por parte del usuario.

La tabla 2 enumera algunas configuraciones de parámetros de acuerdo con una realización de la presente invención.

TABLA 2.

Número de Clics	N (URL)	Delta Promedio (P/N)	1/log (2) + $\log_2$ (P/N)	M (Memoria)
10	680.418	1.273	14	1.159.768
100	80.263	11.650	17	168.853
1000	5.888	55.701.699	29	21.383

La tabla 2 ilustra el uso de memoria para una realización de valores de función hash codificados por codificación Golomb para las URL. La memoria depende tanto del número de las URL (N) como del delta promedio (P/N). La tabla ilustra 3 configuraciones de delta promedio, lo que se corresponde con 14 a 29 bits por URL. Este tipo de compresión hace que sea posible la incorporación de modelos de lenguaje grandes en dispositivos portátiles.

En general, el modelo de lenguaje puede usarse para ayudar a un usuario durante el acceso a la información. Por ejemplo, en un motor de búsqueda, las técnicas de codificación Golomb pueden emplearse para comprobar variantes ortográficas de los términos de búsqueda que proporciona el usuario. En el contexto de un navegador web en un dispositivo portátil, las técnicas de compresiones de codificación Golomb (codificación/ decodificación) pueden adaptarse para comprobar valores de URL alternativos con el fin de corregir unas URL mal escritas.

La figura 8 es un diagrama de flujo simplificado de una realización para la decodificación de una entrada de usuario en relación con un modelo de lenguaje codificado por codificación Golomb. Una entrada de usuario se recibe o se lee, por ejemplo, símbolo a símbolo a partir de un flujo de datos, un archivo, o un dispositivo de entrada (etapa 800). La entrada de usuario se divide en una pluralidad de n–gramas (etapa 802). Cada uno de la pluralidad de n–gramas, se codifica usando una técnica de función hash (etapa 804). Cada n–grama codificado se compara con el modelo de lenguaje codificado por codificación Golomb para identificar las correspondencias posibles (etapa 806). Para cada correspondencia posible se estima estadísticamente una probabilidad de que la correspondencia posible sea una asignación correcta de la entrada de usuario recibida a un elemento dentro del modelo de lenguaje codificado por codificación Golomb (etapa 808). Cualquier número de algoritmos estadísticos puede aplicarse para estimar la probabilidad de que una correspondencia dada sea correcta. En general, cada n–grama puede estar codificado usando una técnica de codificación Golomb, tal como la que se describe anteriormente con respecto a la figura 4.

A pesar de que la presente invención se ha descrito con referencia a unas realizaciones particulares, los expertos en la técnica reconocerán que pueden realizarse cambios en cuanto a la forma y al detalle sin alejarse del alcance de la invención.

REIVINDICACIONES

1. Un procedimiento de compresión de un modelo de lenguaje que comprende:

obtener (300) valores numéricos mediante la aplicación de una función hash a n–gramas de una entrada de usuario y generar una lista de los valores numéricos;
ordenar (302) la lista de valores;
calcular (304) las diferencias entre los valores adyacentes en la lista; y
codificar cada diferencia calculada usando un código Golomb:

elegir un valor M que es aproximadamente igual a la mitad de un valor esperado de las diferencias calculadas entre los valores adyacentes en la lista;
calcular (404, 406) un valor de cociente y un valor de resto a partir de la proporción de la diferencia calculada y el valor M para cada diferencia calculada; y
almacenar el valor de cociente y el valor de resto en una memoria.

2. El procedimiento de la reivindicación 1, en el que la etapa de almacenar comprende:

almacenar el valor de cociente en un formato unario; y
almacenar el valor de resto en un formato binario.

3. El procedimiento de la reivindicación 1, en el que el valor de cociente comprende una proporción de una primera diferencia con respecto al valor M redondeado a la baja hasta el valor entero más cercano.

4. El procedimiento de la reivindicación 1, en el que el valor de resto comprende un resto de una proporción de una primera diferencia con respecto al valor M redondeado a la baja hasta el valor entero más cercano.

5. El procedimiento de la reivindicación 1, que además comprende:

almacenar las diferencias codificadas en una memoria.

6. El procedimiento de la reivindicación 5 en el que las diferencias codificadas ocupan un número de bits (H) en la memoria de acuerdo con una ecuación $$H = N\left[\frac{1}{\log(2)} + \log_2\left(\frac{P}{N}\right)\right]$$

7. El procedimiento de la reivindicación 1, que además comprende:

almacenar instrucciones legibles por ordenador en un medio de almacenamiento, definiendo las instrucciones legibles por ordenador las etapas de generar, ordenar, calcular y codificar.

8. Un sistema para procesar entradas de usuario que comprende:

una interfaz (704) de usuario adaptada para recibir entradas de usuario;
una memoria (706) adaptada para almacenar información y para almacenar un modelo de lenguaje comprimido por codificación Golomb;
un codificador/ decodificador (702) Golomb adaptado para codificar una entrada de usuario y para decodificar elementos del modelo de lenguaje comprimido por codificación Golomb, en el que el codificador/ decodificador Golomb está adaptado para calcular un valor M en base a un valor esperado de las diferencias entre unos valores de función hash adyacentes en una lista ordenada de valores de función hash, estando el codificador/ decodificador Golomb adaptado para calcular un cociente y un resto en base a una proporción de las diferencias con respecto al valor esperado; y
un procesador (710) adaptado para comparar una entrada de usuario frente a elementos del modelo de lenguaje comprimido por codificación Golomb para identificar las correspondencias probables.

9. El sistema de la reivindicación 8, que además comprende:

un conjunto de algoritmos estadísticos adaptados para su uso por el procesador para identificar las correspondencias probables.

10. El sistema de la reivindicación 8, en el que el procesador está adaptado para proporcionar las correspondencias probables identificadas a la interfaz de usuario como salidas para su visualización a un usuario.

11. El sistema de la reivindicación 8, en el que el procesador está adaptado para calcular valores numéricos que se refieren a una entrada de usuario y en el que el codificador/ decodificador Golomb está adaptado para codificar los valores numéricos calculados.

12. Un procedimiento de decodificación de una entrada de usuario con un modelo de lenguaje codificado por codificación Golomb que comprende:

dividir (802) una entrada de usuario recibida en una pluralidad de n–gramas;
codificar (804) cada n–grama usando una técnica de función hash;
comparar (806) cada n–grama codificado con un modelo de lenguaje codificado por codificación Golomb para identificar las correspondencias posibles; y
estimar (808) estadísticamente una probabilidad de que cada correspondencia posible sea una asignación correcta de la entrada de usuario recibida a un elemento dentro del modelo de lenguaje codificado por codificación Golomb;
en el que la etapa de codificar comprende:

obtener valores numéricos mediante la aplicación de una función hash a la pluralidad de n–gramas;
generar una lista ordenada de los valores numéricos;
calcular una diferencia entre los valores adyacentes en la lista;
elegir un valor M para cada diferencia calculada que es aproximadamente igual a la mitad de un valor esperado de la diferencia calculada entre los valores adyacentes en la lista;
calcular un valor de cociente y un valor de resto a partir de la proporción de la diferencia y el valor M para cada diferencia calculada.

13. El procedimiento de la reivindicación 12, en el que la etapa de comparar comprende:

sumar las diferencias entre n–gramas codificados en el modelo de lenguaje hasta que una suma acumulativa es igual a o mayor que un valor del n–grama codificado; y
asociar la entrada de usuario recibida con un valor en el modelo de lenguaje para el n–grama adecuado.

14. El procedimiento de la reivindicación 13, en el que la entrada de usuario recibida asociada comprende la correspondencia posible.

15. El procedimiento de la reivindicación 13, en el que la etapa de codificar comprende además:

combinar el valor de cociente en formato unario y el valor de resto en formato binario para formar el n–grama codificado para cada n–grama.

16. El procedimiento de la reivindicación 12, que además comprende:

almacenar instrucciones legibles por ordenador en un medio de almacenamiento, definiendo las instrucciones legibles por ordenador las etapas de dividir, codificar, comparar, y estimar.

17. Un dispositivo de cálculo portátil adaptado para decodificar una entrada de usuario comparando la entrada de usuario frente a un modelo de lenguaje comprimido por codificación Golomb de acuerdo con el procedimiento de la reivindicación 12.

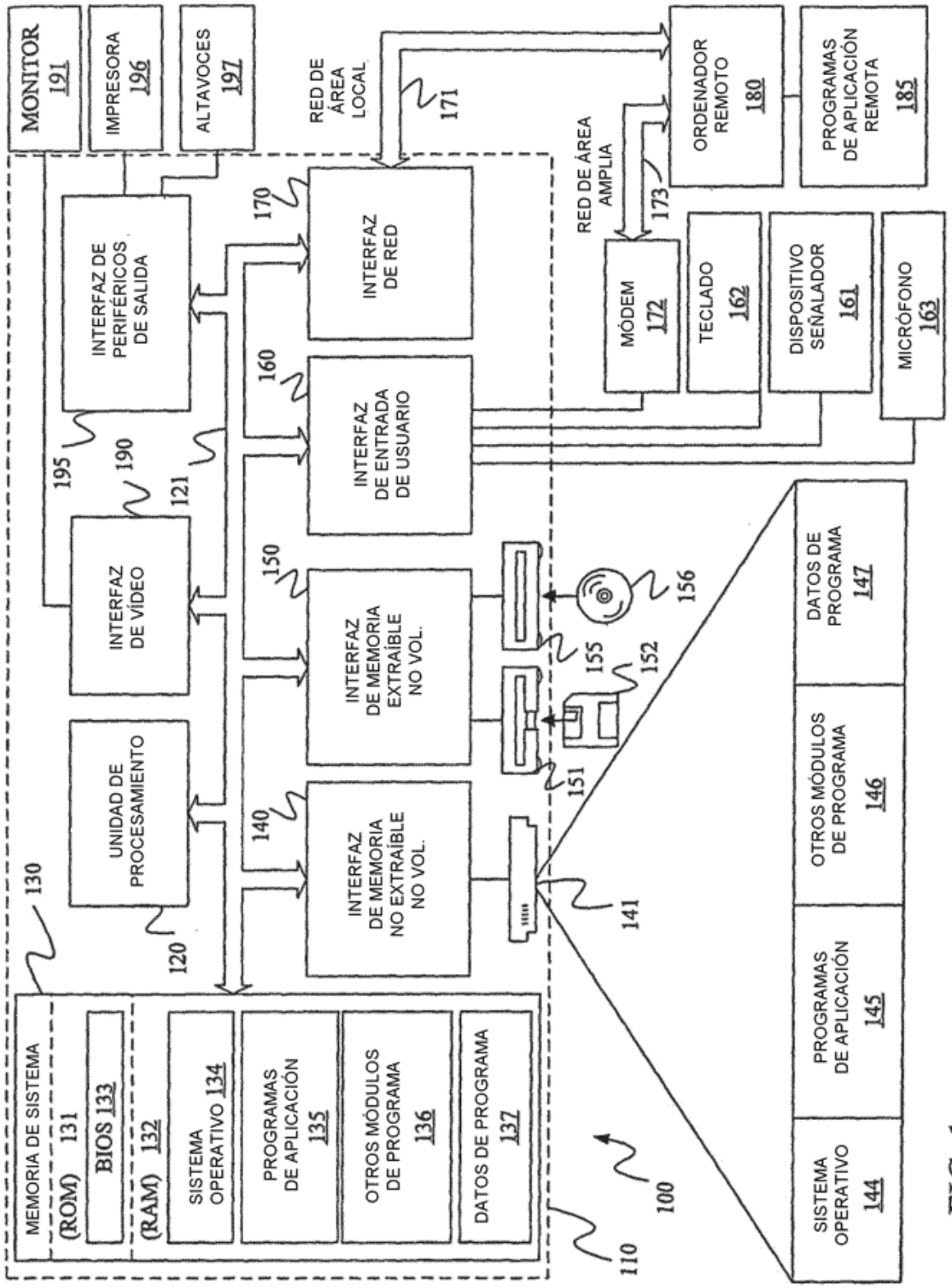
MONITOR 191
IMPRESORA 196
ALTAVOCES 197
RED DE ÁREA LOCAL
171
RED DE ÁREA AMPLIA
173
ORDENADOR REMOTO 180
PROGRAMAS DE APLICACIÓN REMOTA 185
INTERFAZ DE PERIFÉRICOS DE SALIDA
INTERFAZ DE RED
170
MÓDEM 172
TECLADO 162
DISPOSITIVO SEÑALADOR 161
MICRÓFONO 163
160
INTERFAZ DE ENTRADA DE USUARIO
195
190
121
INTERFAZ DE VÍDEO
150
INTERFAZ DE MEMORIA EXTRAÍBLE NO VOL.
156
155
152
151
UNIDAD DE PROCESAMIENTO
140
INTERFAZ DE MEMORIA NO EXTRAÍBLE NO VOL.
141
130
120
MEMORIA DE SISTEMA
(ROM) 131
BIOS 133
(RAM) 132
SISTEMA OPERATIVO 134
PROGRAMAS DE APLICACIÓN 135
OTROS MÓDULOS DE PROGRAMA 136
DATOS DE PROGRAMA 137
110
100
SISTEMA OPERATIVO 144
PROGRAMAS DE APLICACIÓN 145
OTROS MÓDULOS DE PROGRAMA 146
DATOS DE PROGRAMA 147

FIG. 1

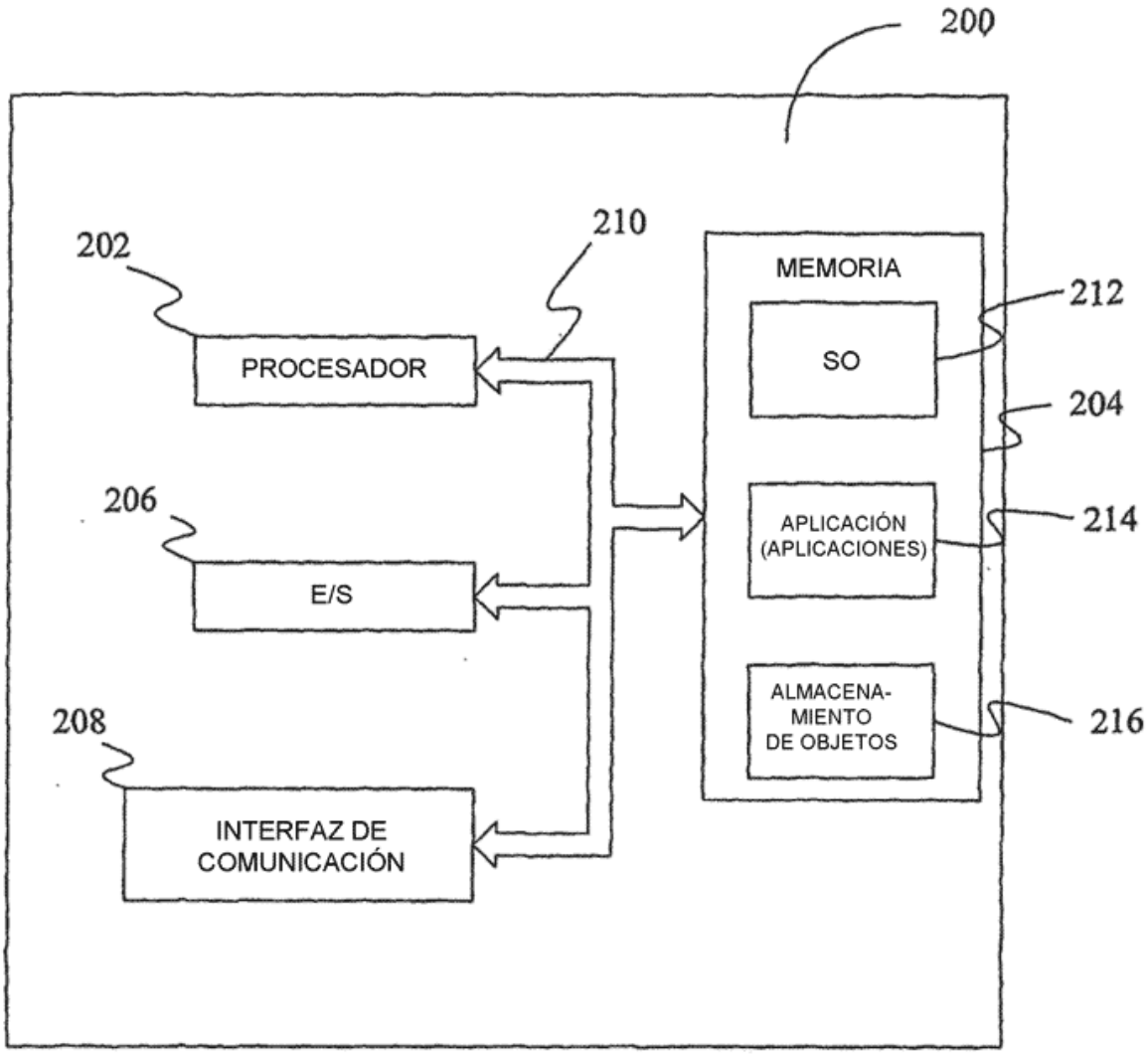
200
202
210
MEMORIA
212
PROCESADOR
SO
204
206
APLICACIÓN
(APLICACIONES)
214
E/S
ALMACENA-
MIENTO
DE OBJETOS
216
208
INTERFAZ DE
COMUNICACIÓN

FIG. 2

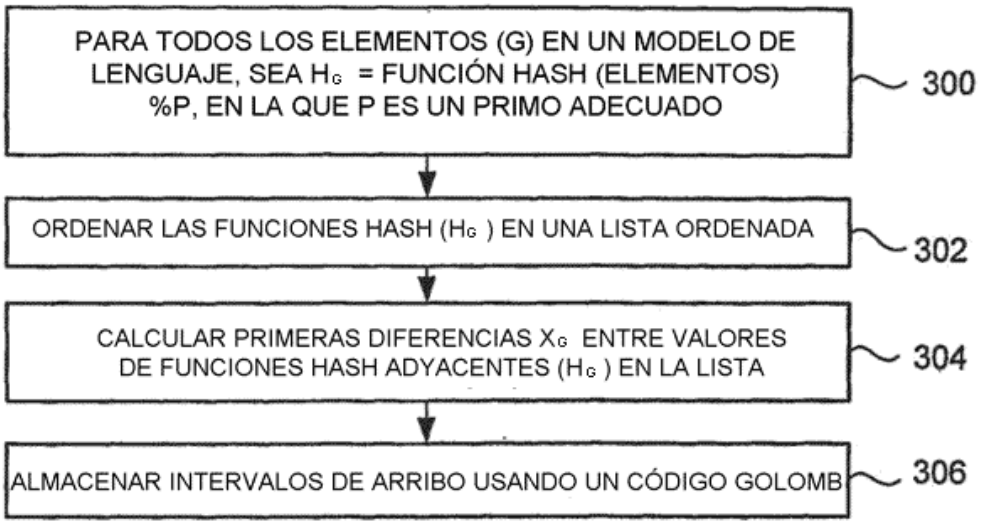
PARA TODOS LOS ELEMENTOS (G) EN UN MODELO DE LENGUAJE, SEA H_G = FUNCIÓN HASH (ELEMENTOS) %P, EN LA QUE P ES UN PRIMO ADECUADO
300
ORDENAR LAS FUNCIONES HASH (H_G) EN UNA LISTA ORDENADA
302
CALCULAR PRIMERAS DIFERENCIAS X_G ENTRE VALORES DE FUNCIONES HASH ADYACENTES (H_G) EN LA LISTA
304
ALMACENAR INTERVALOS DE ARRIBO USANDO UN CÓDIGO GOLOMB
306

FIG. 3

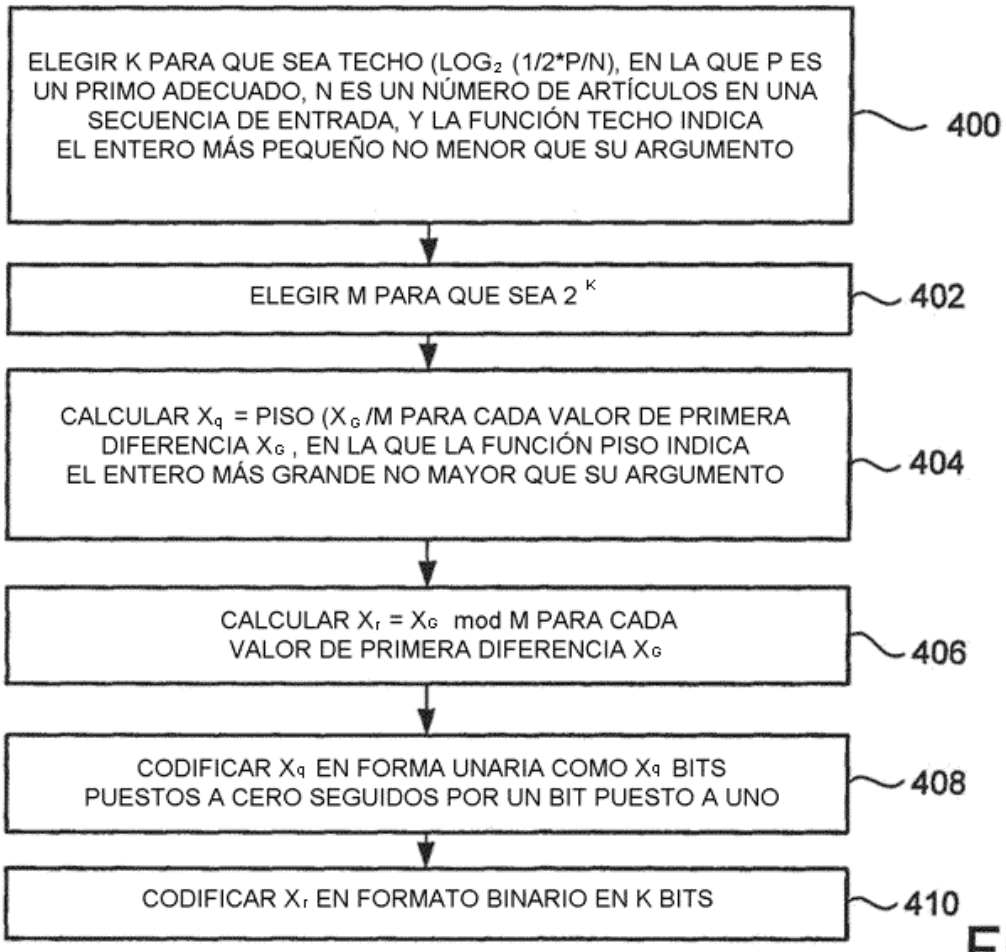
ELEGIR K PARA QUE SEA TECHO (LOG_2 (1/2*P/N), EN LA QUE P ES UN PRIMO ADECUADO, N ES UN NÚMERO DE ARTÍCULOS EN UNA SECUENCIA DE ENTRADA, Y LA FUNCIÓN TECHO INDICA EL ENTERO MÁS PEQUEÑO NO MENOR QUE SU ARGUMENTO
400
ELEGIR M PARA QUE SEA 2^K
402
CALCULAR X_q = PISO (X_G/M PARA CADA VALOR DE PRIMERA DIFERENCIA X_G, EN LA QUE LA FUNCIÓN PISO INDICA EL ENTERO MÁS GRANDE NO MAYOR QUE SU ARGUMENTO
404
CALCULAR $X_r = X_G$ mod M PARA CADA VALOR DE PRIMERA DIFERENCIA X_G
406
CODIFICAR X_q EN FORMA UNARIA COMO X_q BITS PUESTOS A CERO SEGUIDOS POR UN BIT PUESTO A UNO
408
CODIFICAR X_r EN FORMATO BINARIO EN K BITS
410

FIG. 4

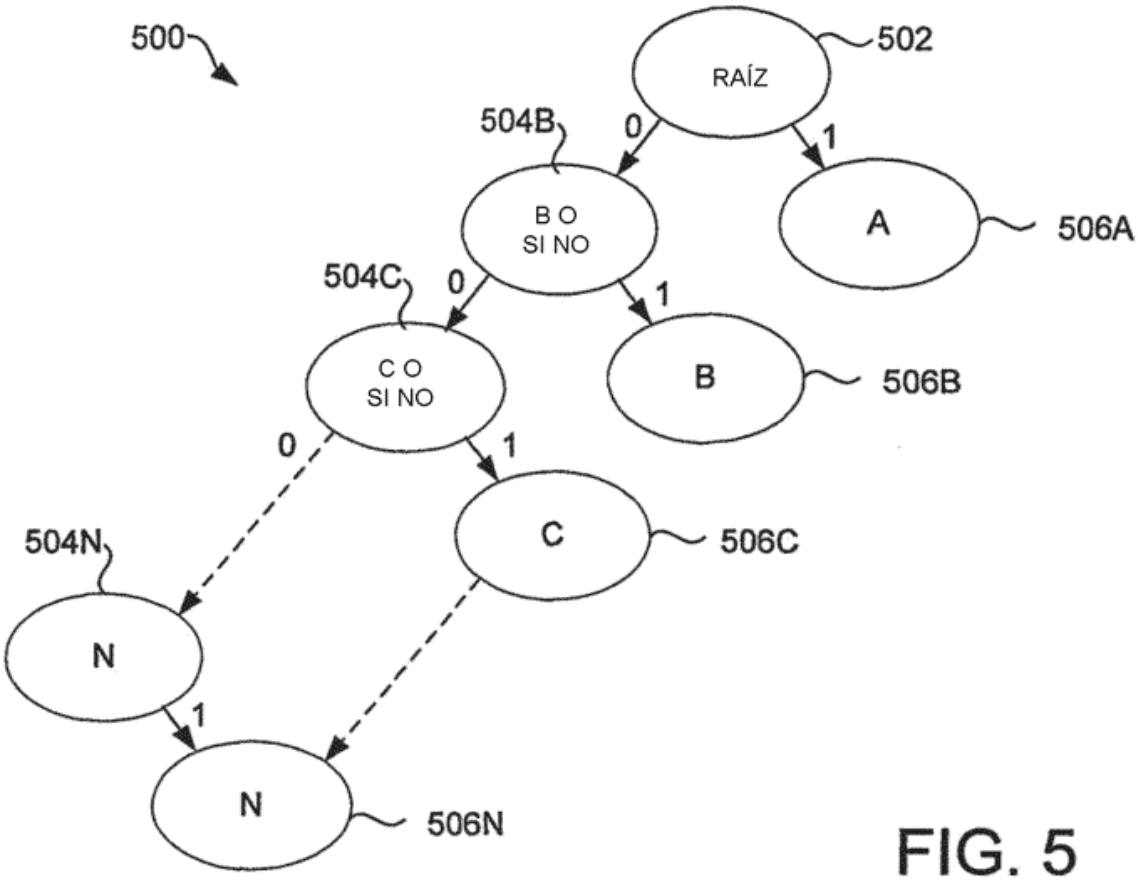
500
RAÍZ
502
504B
0
1
B O
SI NO
A
506A
504C
0
1
C O
SI NO
B
506B
0
1
C
506C
504N
N
1
N
506N

FIG. 5

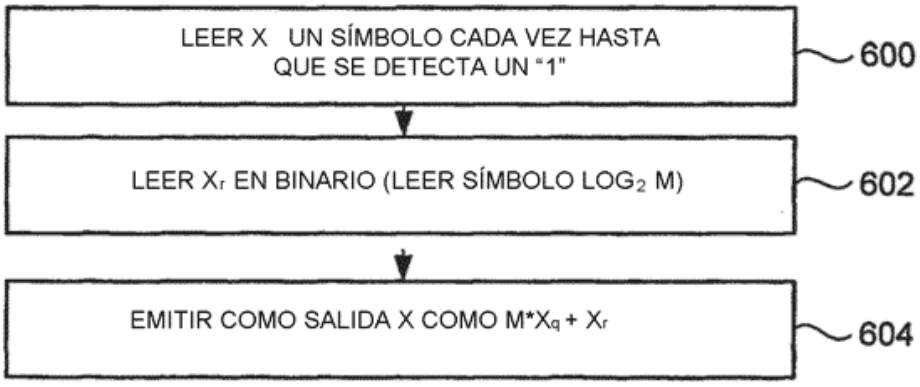
LEER X UN SÍMBOLO CADA VEZ HASTA QUE SE DETECTA UN "1"
600
LEER X_r EN BINARIO (LEER SÍMBOLO LOG_2 M)
602
EMITIR COMO SALIDA X COMO $M*X_q + X_r$
604

FIG. 6

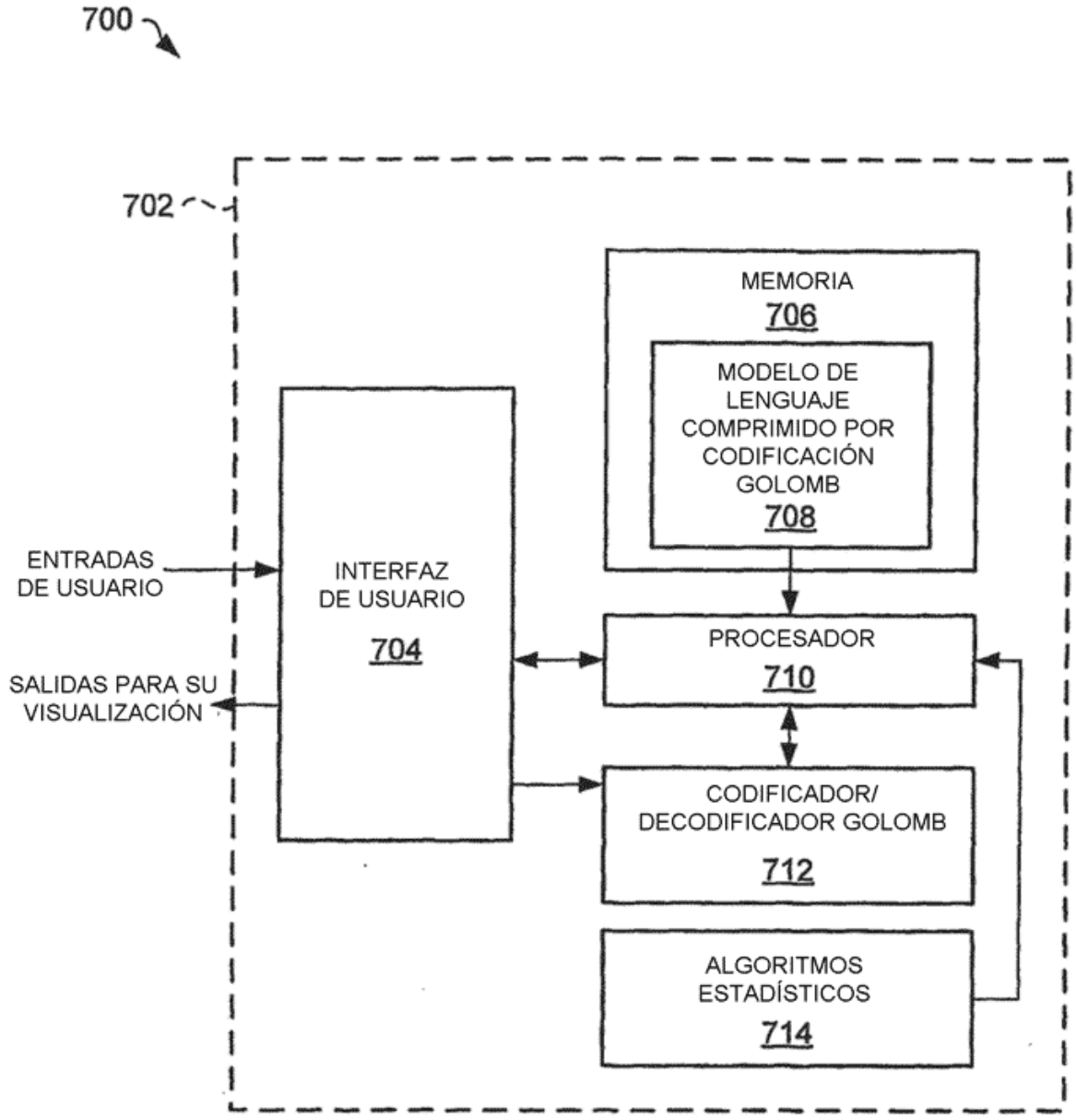
700
702
MEMORIA
706
MODELO DE LENGUAJE COMPRIMIDO POR CODIFICACIÓN GOLOMB
708
ENTRADAS DE USUARIO
INTERFAZ DE USUARIO
704
SALIDAS PARA SU VISUALIZACIÓN
PROCESADOR
710
CODIFICADOR/ DECODIFICADOR GOLOMB
712
ALGORITMOS ESTADÍSTICOS
714

FIG. 7

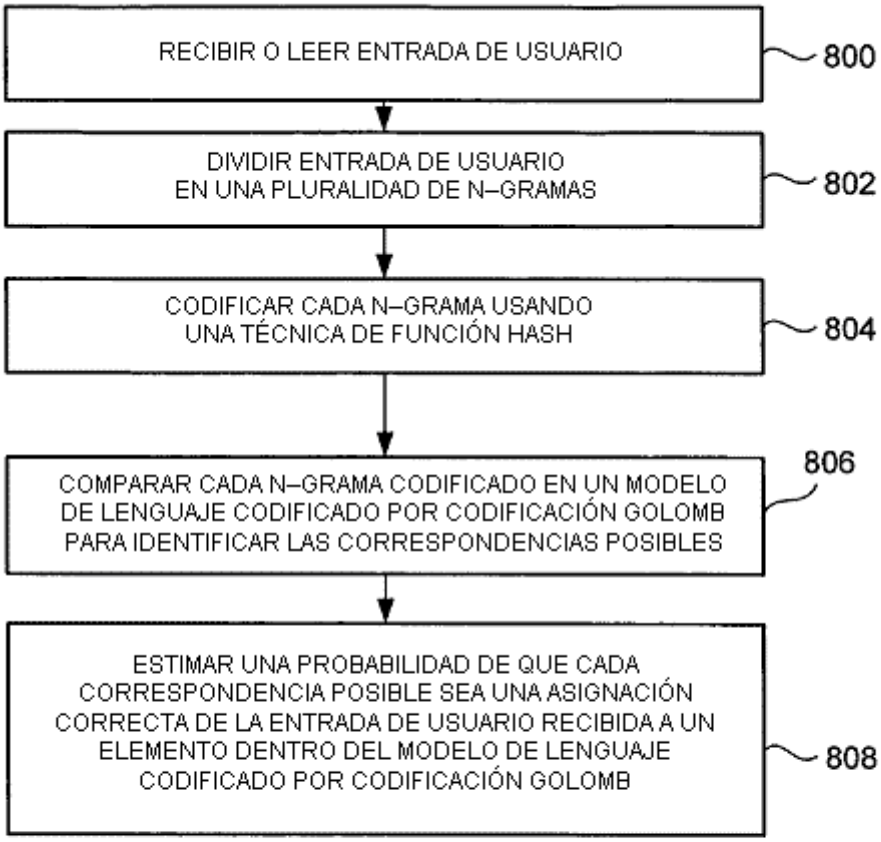
RECIBIR O LEER ENTRADA DE USUARIO
800
DIVIDIR ENTRADA DE USUARIO EN UNA PLURALIDAD DE N–GRAMAS
802
CODIFICAR CADA N–GRAMA USANDO UNA TÉCNICA DE FUNCIÓN HASH
804
COMPARAR CADA N–GRAMA CODIFICADO EN UN MODELO DE LENGUAJE CODIFICADO POR CODIFICACIÓN GOLOMB PARA IDENTIFICAR LAS CORRESPONDENCIAS POSIBLES
806
ESTIMAR UNA PROBABILIDAD DE QUE CADA CORRESPONDENCIA POSIBLE SEA UNA ASIGNACIÓN CORRECTA DE LA ENTRADA DE USUARIO RECIBIDA A UN ELEMENTO DENTRO DEL MODELO DE LENGUAJE CODIFICADO POR CODIFICACIÓN GOLOMB
808

FIG. 8