

MÉTODO Y SISTEMA PARA EL DESCUBRIMIENTO NO SUPERVISADO DE
UNIGRAMAS EN SISTEMAS DE RECONOCIMIENTO DE VOZ

CAMPO DE LA INVENCION

5

La presente invención se relaciona generalmente con el campo del reconocimiento de voz. Más específicamente, la presente invención se relaciona con el descubrimiento no supervisado de unigramas en sistemas de reconocimiento de voz.

10

REIVINDICACIÓN DE PRIORIDAD

Esta solicitud reivindica el beneficio de la solicitud de patente de los EE. UU. núm. 17/520.816, titulada “MÉTODO Y SISTEMA PARA EL DESCUBRIMIENTO NO SUPERVISADO DE UNIGRAMAS EN SISTEMAS DE RECONOCIMIENTO DE VOZ”, presentada en la Oficina de Patentes y Marcas de los Estados Unidos el 08 de noviembre de 2021.

20

ANTECEDENTES DE LA INVENCION

En el campo del análisis y el reconocimiento de voz, los sistemas de reconocimiento de voz continua de gran vocabulario (LVCSR, por sus siglas en inglés) se usan para reconocer a partir de un texto hablado y después aplicar un

25

diccionario de vocabulario o modelo lingüístico de potencialmente 50.000 - 100.000 palabras y frases para producir una transcripción de ese texto hablado.

Por lo tanto, los sistemas de reconocimiento de voz disponibles en la actualidad, como LVCSR, son incapaces, por diseño, de reconocer palabras fuera de vocabulario. Por ejemplo, un sistema LVCSR de un centro de llamadas de una empresa farmacéutica no podrá reconocer términos que describan nombres de medicamentos y afecciones médicas si estos términos están ausentes del modelo lingüístico LVCSR.

Este punto se agrava aún más por el hecho de que los términos específicos de un dominio, que normalmente pueden quedar excluidos de los modelos lingüísticos globales y agnósticos del dominio, también pueden tener una gran importancia en su campo o dominio específico. En relación con el ejemplo de la farmacia: los nombres de los medicamentos, así como las palabras del dominio como “insulina”, “antihistamínico”, “meningitis”, “dolor abdominal”, “antibióticos”, etc., pueden ser muy relevantes en el campo específico de la medicina y la farmacia, pero pueden no estar incluidos en un modelo lingüístico del sistema de análisis de voz y, por lo tanto, no ser reconocidos a partir de voz.

Los sistemas de análisis de voz actualmente disponibles, como LVCSR, pueden mantener un mecanismo de

aprendizaje supervisado que permite aumentar el vocabulario “in situ” añadiendo manualmente frases a un modelo lingüístico basado en n-gramas, en un esfuerzo siempre continuo por actualizar y validar modelos lingüísticos específicos del dominio. Se puede apreciar que este enfoque requiere una amplia intervención humana.

RESUMEN DE LA INVENCION

Por lo tanto, se requiere un método y un sistema para el descubrimiento automático, continuo y no supervisado de unigramas en un sistema de reconocimiento de voz.

Las modalidades de la invención pueden incluir un método de descubrimiento automático de unigramas en un elemento de datos de voz, mediante al menos un procesador.

Las modalidades del método pueden incluir recibir un modelo lingüístico, que puede incluir una pluralidad de n-gramas, donde cada n-grama puede incluir uno o más unigramas; aplicar un modelo acústico de aprendizaje automático (ML) en uno o más primeros elementos de datos de voz para obtener una función de distribución de caracteres; aplicar un decodificador voraz en la función de distribución de caracteres, para predecir un corpus inicial de unigramas; filtrar uno o más unigramas del corpus inicial para obtener un corpus de unigramas candidatos que

no estén incluidos en el modelo lingüístico; analizar uno o más primeros elementos de datos de voz para extraer al menos un n-grama que incluye un unigrama candidato; y actualizar el modelo lingüístico para incluir el al menos
5 un n-grama extraído.

De acuerdo con algunas modalidades, el al menos un procesador puede aplicar un decodificador de haz en un segundo elemento de datos de voz (p. ej., durante una etapa de inferencia), para producir al menos una transcripción
10 del segundo elemento de datos de voz, basado en el modelo lingüístico actualizado.

De acuerdo con algunas modalidades, la función de distribución de caracteres puede representar una probabilidad de aparición de uno o más caracteres
15 lingüísticos en el uno o más primeros elementos de datos de voz.

De acuerdo con algunas modalidades, el al menos un procesador puede volver a entrenar el modelo acústico de ML, basado en el al menos un segundo elemento de datos
20 de voz, usando el al menos un n-grama extraído como datos de supervisión.

De acuerdo con algunas modalidades, el al menos un procesador puede filtrar uno o más unigramas mediante:
(a) para uno o más unigramas candidatos, calcular una
25 probabilidad de error ortográfico, que representa una probabilidad de que un unigrama relevante sea una versión

con error ortográfico de un unigrama que ya está incluido en el modelo lingüístico; y (b) filtrar los unigramas candidatos que corresponden a una probabilidad de error ortográfico que excede un umbral predefinido.

5 De acuerdo con algunas modalidades, el al menos un procesador puede calcular una probabilidad de error ortográfico mediante: calcular un valor de distancia Levenshtein entre el unigrama candidato y al menos un segundo unigrama, ya incluido en el modelo lingüístico; 10 calcular una puntuación de frecuencia, que representa una relación de apariencias entre el unigrama candidato y el al menos un segundo unigrama en el uno o más primeros elementos de datos de voz; y calcular la probabilidad de error ortográfico basado en el valor de distancia de 15 Levenshtein y la puntuación de frecuencia.

De acuerdo con algunas modalidades, el decodificador voraz puede adaptarse para emitir, para cada unigrama del corpus inicial, un nivel de confianza respectivo. En tales modalidades, el al menos un 20 procesador puede configurarse para calcular una probabilidad de error ortográfico adicionalmente mediante: el cálculo de una puntuación de confianza, que representa un promedio del nivel de confianza para una o más apariciones del unigrama candidato en el uno o más 25 primeros elementos de datos de voz; y el cálculo de la

probabilidad de error ortográfico basado además en la puntuación de confianza.

De acuerdo con algunas modalidades, el modelo lingüístico puede incluir además una definición de una o
5 más reglas sintácticas del lenguaje. En tales modalidades, el al menos un procesador puede configurarse para calcular la distancia de Levenshtein mediante: el cálculo de un número de ediciones de un solo carácter entre el unigrama candidato y el segundo unigrama; y el cálculo del valor de
10 la distancia de Levenshtein basado en una o más reglas sintácticas del lenguaje y el número de ediciones de un solo carácter.

De acuerdo con algunas modalidades, el al menos un procesador puede calcular, para uno o más unigramas
15 candidatos, una probabilidad de espacio faltante. La probabilidad de espacio faltante puede representar una probabilidad de que el unigrama candidato pueda ser una concatenación de dos unigramas que ya están incluidos en el modelo lingüístico. Además, el al menos un procesador puede
20 filtrar los unigramas candidatos que corresponden a una probabilidad de espacio faltante que excede un umbral predefinido.

De acuerdo con algunas modalidades, el al menos un procesador puede: calcular, para al menos un primer
25 unigrama candidato, un primer vector de incrustación de unigrama, basado en el corpus de unigramas candidatos;

calcular, para al menos un segundo unigrama candidato, un segundo vector de incrustación de unigrama, basado en el corpus de unigramas candidatos; calcular una puntuación de similitud basada en el primer vector de incrustación de unigramas y el segundo vector de incrustación de unigramas, y calcular además la probabilidad de error ortográfico basándose en la puntuación de similitud.

De acuerdo con algunas modalidades, el al menos un procesador puede recibir un corpus de documentos, que incluye una pluralidad de documentos, donde cada documento puede asociarse con un dominio de tema específico, y donde cada documento incluye una pluralidad de unigramas de documentos.

El al menos un procesador puede calcular, para uno o más unigramas candidatos, una puntuación del dominio basada en la pluralidad de unigramas del documento, en donde dicha puntuación del dominio representa una probabilidad de que el unigrama candidato pueda ser pertinente para al menos un dominio específico.

Adicionalmente, para uno o más unigramas candidatos, el al menos un procesador puede: compilar una lista de contexto que puede incluir un subconjunto de unigramas de documentos. El subconjunto de unigramas de documentos puede (a) tener una puntuación del dominio que excede un umbral predefinido y (b) aparecer en uno o más primeros elementos de datos de voz. Para cada documento,

el al menos un procesador puede obtener un grupo de intersección que puede incluir unigramas que aparecen en el documento y en la lista de contexto. Para cada documento, el al menos un procesador puede calcular una
5 puntuación de corrección que representa la relevancia del unigrama candidato para el documento, basándose en las puntuaciones del dominio de los unigramas de documentos en el grupo de intersección. El al menos un procesador puede filtrar posteriormente u omitir unigramas
10 candidatos que corresponden a una puntuación de corrección máxima que está por debajo de un umbral predefinido.

De acuerdo con algunas modalidades, el decodificador voraz puede adaptarse para emitir, para cada
15 unigrama del corpus inicial, un nivel de confianza respectivo. En tales modalidades, el al menos un procesador puede analizar un elemento de datos de voz del uno o más primeros elementos de datos de voz mediante: para uno o más unigramas candidatos, localizar en el elemento de datos de
20 voz, un n-grama de unigramas adyacentes, puede incluir el unigrama candidato; si (a) los unigramas de dicho n-grama corresponden a un nivel de confianza más allá de un valor predefinido, y (b) dicho n-grama incluye más de un número umbral predefinido de unigramas entonces el modelo
25 lingüístico puede actualizarse para incluir al menos un n-grama extraído. De cualquier otra manera, el unigrama

candidato puede filtrarse del corpus de unigramas candidatos.

Las modalidades de la invención pueden incluir un sistema para descubrir automáticamente unigramas en un
5 elemento de datos de voz. Las modalidades del sistema pueden incluir: un dispositivo de memoria no transitorio, en donde pueden almacenarse módulos del código de instrucciones, y al menos un procesador asociado con el dispositivo de memoria, y configurado para ejecutar los
10 módulos del código de instrucciones.

Tras la ejecución de los módulos de código de instrucciones, el al menos un procesador puede configurarse para: recibir un modelo lingüístico, puede incluir una pluralidad de n-gramas, cada uno puede incluir uno o más
15 unigramas; aplicar un modelo acústico de ML en uno o más primeros elementos de datos de voz para obtener una función de distribución de caracteres; aplicar un decodificador voraz en la función de distribución de caracteres, para predecir un corpus inicial de unigramas; filtrar uno o más
20 unigramas del corpus inicial para obtener un corpus de unigramas candidatos, dichos unigramas candidatos no incluidos en el modelo lingüístico; analizar el uno o más primeros elementos de datos de voz, para extraer al menos un n-grama que puede incluir un unigrama candidato; y
25 actualizar el modelo lingüístico para incluir al menos un n-grama extraído.

Adicionalmente, las modalidades del sistema pueden incluir un decodificador de haz, configurado para recibir al menos un segundo elemento de datos de voz y producir una transcripción del al menos un segundo elemento de datos de voz, basado en el modelo lingüístico actualizado.

BREVE DESCRIPCIÓN DE LAS FIGURAS

La materia considerada como la invención se señala particularmente y se reivindica claramente en la parte final de la especificación. Sin embargo, la invención, tanto en cuanto a organización como método de operación, junto con los objetos, características y ventajas de esta, se puede entender mejor haciendo referencia a la siguiente descripción detallada cuando se lea con los dibujos adjuntos en los que:

La Figura 1 es un diagrama en bloque, que ilustra un dispositivo informático que puede incluirse en un sistema de reconocimiento de voz, de acuerdo con algunas modalidades de la invención;

la Figura 2 es un diagrama en bloque, que ilustra una descripción general de un sistema para el reconocimiento de voz, de acuerdo con algunas modalidades de la invención;

la Figura 3 es un diagrama en bloque, que ilustra el flujo de datos en un sistema para el reconocimiento de voz, de acuerdo con algunas modalidades de la invención;

la Figura 4 es un diagrama en bloque, que ilustra otra vista de un sistema para el reconocimiento de voz, de acuerdo con algunas modalidades de la invención;

la Figura 5 es un diagrama en bloque, que ilustra un ejemplo de un módulo de extracción de unigrama, que puede incluirse en un sistema para el reconocimiento de voz, de acuerdo con algunas modalidades de la invención;

la Figura 6 es un mapa de calor que ilustra la probabilidad de sustitución y supresión de letras del alfabeto inglés en un sistema de reconocimiento de voz, de acuerdo con algunas modalidades de la invención;

la Figura 7 es un diagrama en bloque, que ilustra otro ejemplo de un módulo de extracción de unigrama, que puede incluirse en un sistema para el reconocimiento de voz, de acuerdo con algunas modalidades de la invención; y

la Figura 8 es un diagrama de flujo, que ilustra un método para descubrir automáticamente unigramas en un elemento de datos de voz mediante un sistema para el reconocimiento de voz, de acuerdo con algunas modalidades de la invención.

Se entenderá que, por simplicidad y claridad de la ilustración, los elementos mostrados en las figuras no se han dibujado necesariamente a escala. Por ejemplo, las dimensiones de algunos de los elementos pueden estar
5 exageradas con respecto a otros elementos para mayor claridad. Además, cuando se considere adecuado, los números de referencia pueden repetirse entre las figuras para indicar elementos correspondientes o análogos.

DESCRIPCIÓN DETALLADA DE LA PRESENTE INVENCION

Un experto en la técnica reconocerá que la invención puede incorporarse en otras formas específicas sin apartarse del espíritu o características esenciales de esta. Por lo tanto, las modalidades anteriores deben considerarse ilustrativas en todos los aspectos en lugar de limitar la invención descrita en la presente descripción. El alcance de la invención se indica, por lo tanto, mediante las reivindicaciones adjuntas, más que por la descripción anterior, y todos los cambios que entran dentro del significado y se pretende, por lo tanto, que el rango de equivalencia de las reivindicaciones se incluya en estas.

En la siguiente descripción detallada, se exponen numerosos detalles específicos para proporcionar una comprensión completa de la invención. Sin embargo, los expertos en la técnica entenderán que la presente invención puede practicarse sin estos detalles específicos. En otras instancias, los métodos, procedimientos y componentes muy conocidos no se han descrito en detalle para no oscurecer la presente invención. Algunas características o elementos descritos con respecto a una modalidad pueden combinarse con características o elementos descritos con respecto a otras modalidades. En aras de la claridad, puede que no

se repita un análisis de características o elementos iguales o similares.

Aunque las modalidades de la invención no se limitan en este sentido, los análisis que usan términos
5 tales como, por ejemplo, “que procesa”, “que computa”, “que calcula”, “que determina”, “que establece”, “que analiza”, “que verifica” o lo similar, pueden referirse a operaciones y/o procesos de una computadora, una plataforma informática, un sistema informático u otro
10 dispositivo electrónico informático, que manipula y/o transforma datos representados como cantidades físicas (p. ej., electrónicas) dentro de los registros y/o memorias de la computadora en otros datos representados de manera similar como cantidades físicas dentro de los
15 registros y/o memorias de la computadora u otro medio de almacenamiento no transitorio de información que puede almacenar instrucciones para realizar operaciones y/o procesos.

Aunque las modalidades de la invención no están
20 limitadas en este sentido, los términos “pluralidad” y “una pluralidad” como se usan en la presente descripción pueden incluir, por ejemplo, “múltiples” o “dos o más”. Los términos “pluralidad” o “una pluralidad” se pueden usar en toda la especificación para describir dos o más
25 componentes, dispositivos, elementos, unidades,

parámetros o lo similar. El término “conjunto”, cuando se usa en la presente descripción, puede incluir uno o más elementos.

A menos que se indique explícitamente, las
5 modalidades del método descritas en la presente descripción no se limitan a un orden o secuencia particular. Adicionalmente, algunas de las modalidades del método descrito o elementos de este pueden ocurrir o realizarse simultáneamente, en el mismo punto en el
10 tiempo, o concurrentemente.

La siguiente Tabla 1 incluye un glosario de términos usados en la siguiente descripción.

Tabla 1

Voz, elemento de datos de voz	Los términos “voz” y “elemento de datos de voz” pueden usarse en la presente descripción indistintamente para indicar un elemento de datos tal como un archivo de audio, o un archivo audiovisual, o datos de transmisión, que pueden incluir una grabación, o una representación de voz humana en un lenguaje natural (p. ej., inglés, francés, etc.).
Reconocimiento de voz, transcripción de voz	Los términos “reconocimiento de voz” y “transcripción de voz” pueden usarse en la presente descripción indistintamente para indicar un proceso automatizado para recibir un elemento de datos de audio o audiovisual que contiene habla humana, y extraer de este un elemento de datos de texto que representa una transcripción de voz.
Unigrama	El término “unigrama” puede usarse en la presente descripción para referirse a un elemento, tal como una palabra, que puede

	incluirse o extraerse de una muestra dada de texto o habla.
Ngrama (n-grama)	El término “n-grama” puede usarse en la presente descripción para referirse a una secuencia contigua de un número (“n”, p. ej., 2, 3, etc.) de unigramas (p. ej., palabras) que pueden incluirse o extraerse de una muestra dada de texto o habla.
Modelo acústico, Red neuronal acústica, Función de distribución de unigrama	Los términos “modelo acústico” y “red neuronal acústica” pueden usarse en la presente descripción indistintamente para indicar un paradigma de procesamiento de información de audio. Como se conoce en la técnica, un modelo acústico puede adaptarse para recibir un elemento de datos de voz y extraer de este una función de distribución de caracteres. La función de distribución de caracteres puede representar una probabilidad de aparición de cada carácter (p. ej., letra) en el elemento de datos de voz subyacente.
Decodificador voraz	El término “decodificador voraz” puede usarse en la presente descripción para referirse a un proceso de búsqueda automatizado, que puede extraer al menos un n-grama de un elemento de datos de voz dado. Como se conoce en la técnica, un proceso de decodificación voraz puede recibir una función de distribución de caracteres (p. ej., de un modelo acústico) correspondiente a un elemento de datos de voz subyacente. La función de distribución de caracteres puede ser una secuencia de distribuciones de caracteres (p. ej., una distribución de caracteres por uno o más fotogramas de audio de normalmente 25 milisegundos). Cada distribución de caracteres puede ser un vector concreto de números (p. ej., una realización de una función de distribución). El proceso de decodificación voraz puede entonces extraer o producir un n-grama que incluya unigramas correspondientes a los valores máximos de la función de distribución de caracteres.
Decodificador de haz	El término “decodificador de haz” puede usarse en la presente descripción para referirse a otro proceso de búsqueda

	automatizado, que puede extraer al menos un n-grama de un elemento de datos de voz dado. Como se conoce en la técnica, un decodificador de haz puede recibir una función de distribución de caracteres (p. ej., de un modelo acústico) correspondiente a un elemento de datos de voz subyacente. El proceso de decodificación de haz puede entonces aplicar un modelo lingüístico basado en n-gramas para extraer o producir al menos un n-grama que (a) corresponda a valores de función de distribución de caracteres altos y (b) se incluye en el modelo lingüístico basado en n-gramas.
Vocabulario, modelo lingüístico	Los términos “vocabulario” y “modelo lingüístico” pueden usarse en la presente descripción indistintamente para referirse a un modelo estadístico, comúnmente usado en aplicaciones de procesamiento del lenguaje natural (NLP, por sus siglas en inglés), para determinar la probabilidad de que una secuencia dada de palabras ocurra en una oración de un lenguaje natural, tal como el inglés o el español.
Red neuronal	El término “red neuronal” (NN, por sus siglas en inglés) o “red neuronal artificial” (ANN, por sus siglas en inglés), p. ej., una red neuronal que implementa una función de aprendizaje automático (ML) o inteligencia artificial (AI, por sus siglas en inglés), puede referirse a un paradigma de procesamiento de información que puede incluir nodos, denominados neuronas, organizadas en capas, con enlaces entre las neuronas. Los enlaces pueden transferir señales entre neuronas y pueden asociarse con ponderaciones. Una NN puede configurarse o entrenarse para una tarea específica, p. ej., reconocimiento o clasificación de patrones. El entrenamiento de la NN para la tarea específica puede implicar afinar estas ponderaciones basado en ejemplos. Cada neurona de una capa intermedia o última puede recibir una señal de entrada, por ejemplo, una suma ponderada de señales de salida de otras neuronas, y puede procesar la señal de entrada usando una función lineal o no lineal (p. ej., una

	función de activación). Los resultados de las capas de entrada e intermedias pueden transferirse a otras neuronas y los resultados de la capa de salida pueden proporcionarse como la salida de la NN. Típicamente, las neuronas y los enlaces dentro de una NN se representan mediante constructos matemáticos, tales como funciones de activación y matrices de elementos de datos y ponderaciones. Un procesador, p. ej., CPU o unidades de procesamiento de gráficos (GPU, por sus siglas en inglés), o un dispositivo de hardware dedicado, puede realizar los cálculos relevantes.
--	---

Ahora se hace referencia a la Figura 1, que es un diagrama en bloque que ilustra un dispositivo informático, que puede incluirse dentro de un sistema para el

5 reconocimiento de voz, de acuerdo con algunas modalidades.

El dispositivo informático 1 puede incluir un procesador o controlador 2 que puede ser, por ejemplo, un procesador de la unidad de procesamiento central (CPU, por sus siglas en inglés), un chip o cualquier computadora

10 adecuada o dispositivo computacional, un sistema operativo 3, una memoria 4, código ejecutable 5, un sistema de almacenamiento 6, dispositivos de entrada 7 y dispositivos de salida 8. El procesador 2 (o uno o más controladores o procesadores, posiblemente a través de múltiples unidades o

15 dispositivos) puede configurarse para realizar los métodos descritos en la presente descripción, y/o ejecutar o actuar como los diversos módulos, unidades, etc. Más de un dispositivo informático 1 puede incluirse en, y uno o más

dispositivos informáticos 1 pueden actuar como componentes de, un sistema de acuerdo con las modalidades de la invención.

El sistema operativo 3 puede ser o puede incluir cualquier segmento de código (p. ej., uno similar al código ejecutable 5 descrito en la presente descripción) diseñado y/o configurado para realizar tareas que involucren la coordinación, planificación, arbitraje, supervisión, control o de cualquier otra manera gestión del funcionamiento del dispositivo informático 1, por ejemplo, programación de programas o tareas del software o permitir la comunicación de programas de software u otros módulos o unidades. El sistema operativo 3 puede ser un sistema operativo comercial. Se observará que un sistema operativo 3 puede ser un componente opcional, por ejemplo, en algunas modalidades, un sistema puede incluir un dispositivo informático que no requiere o incluye un sistema operativo 3.

La memoria 4 puede ser o puede incluir, por ejemplo, una memoria de acceso aleatorio (RAM, por sus siglas en inglés), una memoria de solo lectura (ROM, por sus siglas en inglés), una RAM dinámica (DRAM, por sus siglas en inglés), una DRAM sincrónica (SD-RAM), un chip de memoria de tasa de datos doble (DDR, por sus siglas en inglés), una memoria flash, una memoria volátil, una memoria no volátil, una memoria caché, un regulador, una

unidad de memoria a corto plazo, una unidad de memoria a largo plazo, u otras unidades de memoria o unidades de almacenamiento adecuadas. La memoria 4 puede ser o puede incluir una pluralidad de unidades de memoria posiblemente diferentes. La memoria 4 puede ser un medio legible por computadora o procesador no transitorio, o un medio de almacenamiento no transitorio informático, por ejemplo, una RAM. En una modalidad, un medio de almacenamiento no transitorio tal como la memoria 4, una unidad de disco duro, otro dispositivo de almacenamiento, etc., puede almacenar instrucciones o código que cuando se ejecutan mediante un procesador puede hacer que el procesador realice métodos como se describe en la presente descripción.

El código ejecutable 5 puede ser cualquier código ejecutable, p. ej., una aplicación, un programa, un proceso, tarea o secuencia. El código ejecutable 5 puede ejecutarse por el procesador o el controlador 2 posiblemente bajo el control del sistema operativo 3. Por ejemplo, el código ejecutable 5 puede ser una aplicación que puede realizar el reconocimiento de voz como se describe más adelante en la presente descripción. Aunque, en aras de la claridad, un único elemento del código ejecutable 5 se muestra en la Figura 1, un sistema de acuerdo con algunas modalidades de la invención puede incluir una pluralidad de segmentos de código ejecutable

similares al código ejecutable 5 que puede cargarse en la memoria 4 y hacer que el procesador 2 realice los métodos descritos en la presente.

El sistema de almacenamiento 6 puede ser o puede
5 incluir, por ejemplo, una memoria Flash como se conoce en la técnica, una memoria que es interna a, o incrustada en, un microcontrolador o chip como se conoce en la técnica, una unidad de disco duro, una unidad de accionamiento (CD-R, por sus siglas en inglés) que se
10 extiende en CD, un disco Blu-ray (BD, por sus siglas en inglés), un dispositivo de bus serie universal (USB, por sus siglas en inglés) u otra unidad de almacenamiento extraíble y/o fija adecuada. Los datos pertenecientes a uno o más elementos de datos de voz pueden almacenarse en
15 el sistema de almacenamiento 6 y pueden cargarse desde el sistema de almacenamiento 6 a la memoria 4, donde pueden ser procesados por el procesador o controlador 2. En algunas modalidades, algunos de los componentes mostrados en la Figura 1 pueden omitirse. Por ejemplo, la memoria 4
20 puede ser una memoria no volátil que tiene la capacidad de almacenamiento del sistema de almacenamiento 6. En consecuencia, aunque se muestra como un componente separado, el sistema de almacenamiento 6 puede estar incrustado o incluido en la memoria 4.

25 Los dispositivos de entrada 7 pueden ser o pueden incluir cualquier dispositivo, componente o

sistema de entrada adecuado, p. ej., un teclado o bloque de teclas desmontable, un ratón y lo similar. Los dispositivos de salida 8 pueden incluir uno o más pantallas o monitores, altavoces y/o cualquier otro dispositivo de salida adecuado (posiblemente desmontables). Cualquier dispositivo de entrada/salida (E/S) aplicable puede conectarse al dispositivo informático 1 como se muestra por los bloques 7 y 8. Por ejemplo, una tarjeta de interfaz de red (NIC, por sus siglas en inglés) cableada o inalámbrica, un dispositivo de bus serie universal (USB) o un disco duro externo se puede incluir en los dispositivos de entrada 7 y/o dispositivos de salida 8. Se reconocerá que cualquier número adecuado de dispositivos de entrada 7 y dispositivo de salida 8 puede conectarse operativamente al dispositivo informático 1 como se muestra en los bloques 7 y 8.

Un sistema de acuerdo con algunas modalidades de la invención puede incluir componentes tales como, entre otros, una pluralidad de unidades centrales de procesamiento (CPU) o cualquier otro procesador o controlador de propósito múltiple adecuado o específico (p. ej., similar al elemento 2), una pluralidad de unidades de entrada, una pluralidad de unidades de salida, una pluralidad de unidades de memoria y una pluralidad de unidades de almacenamiento.

En los últimos años, un nuevo enfoque del reconocimiento de voz, comúnmente denominado “reconocimiento de extremo a extremo”, ha abierto otra opción para detectar nuevas palabras. A diferencia de los
5 anteriores métodos de reconocimiento de voz, que se basan en un grafo a nivel de palabra para realizar el reconocimiento, en el que cada palabra se construye a partir de una secuencia de fonemas, el reconocimiento de extremo a extremo ha cambiado completamente al nivel de
10 caracteres. Esto significa que, en lugar de una salida que es una secuencia (o grafo, “red”) de palabras o fonemas, el reconocimiento de voz de extremo a extremo genera una función de distribución de caracteres.

Se hace referencia ahora a la Figura 2, que es
15 un diagrama en bloque, que ilustra una visión general de un sistema 100 para el reconocimiento de voz, de acuerdo con algunas modalidades de la invención. Como se explica en la presente descripción, el sistema 100 puede configurarse para descubrir automáticamente unigramas (p.
20 ej., palabras) en uno o más elementos de datos de voz, para mejorar un modelo de lenguaje dado 160A, y después utilizar el modelo de lenguaje mejorado 160A con un decodificador de haz para realizar el reconocimiento de voz en una o más nuevas muestras de elementos de datos de
25 voz.

Como se ilustra en la Figura 2, el sistema 100 puede implementar el reconocimiento de voz de extremo a extremo, mientras aprovecha una sinergia entre tres funciones separadas (marcadas por líneas discontinuas):
5 análisis acústico, decodificación voraz y decodificación de haz, como se elabora en la presente descripción.

De acuerdo con algunas modalidades de la invención, el sistema 100 puede implementarse como un módulo de software, un módulo de hardware o cualquier
10 combinación de estos. Por ejemplo, el sistema 100 puede ser o puede incluir un dispositivo informático tal como el elemento 1 de la Figura 1, y puede adaptarse para ejecutar uno o más módulos del código ejecutable (p. ej., elemento 5 de la Figura 1) para realizar el
15 reconocimiento de voz, como se describe más adelante en la presente descripción.

Como se muestra en la Figura 2, las flechas pueden representar el flujo de uno o más elementos de datos hacia y desde el sistema 100 y/o entre los módulos
20 o elementos del sistema 100. Algunas flechas se han omitido en la Figura 2 para fines de claridad.

Como se muestra en la Figura 2, el sistema 100 puede recibir (p. ej., desde el dispositivo de entrada 7 de la Figura 1) una versión inicial de un modelo
25 lingüístico 160A. De acuerdo con algunas modalidades, el modelo lingüístico 160A puede incluir una pluralidad de

n-gramas, donde cada n-grama puede incluir uno o más unigramas. Por ejemplo, como se conoce en la técnica del procesamiento del lenguaje natural (PNL), el modelo lingüístico 160A basado en n-gramas puede incluir una pluralidad de n-gramas, cada uno representando un conjunto o combinación de unigramas (p. ej., palabras). Cada n-grama puede asociarse con una probabilidad de n-grama, representando probabilidad o frecuencia de aparición de la respectiva combinación de unigramas en un corpus de documentos. Por ejemplo, una probabilidad de n-grama del n-grama “Te amo” puede ser mayor que una probabilidad de n-grama del n-grama “Me encantan los panqueques”, lo que representa una mayor probabilidad de que el unigrama “te” aparezca en un elemento de datos de voz 20 después del n-grama “Me encanta”.

De acuerdo con algunas modalidades, el sistema 100 puede recibir (p. ej., mediante el dispositivo de entrada 7 de la Figura 1) uno o más elementos de datos de audio o habla 20. El uno o más elementos de datos de audio o habla 20 pueden ser, por ejemplo, un archivo de audio o un flujo de datos que puede incluir o puede representar el habla humana en un lenguaje natural (p. ej., inglés, francés, etc.). Como se elabora en la presente descripción, el sistema 100 puede configurarse para realizar un descubrimiento no supervisado de unigramas en el uno o más

elementos de datos de voz 20, y posteriormente producir un elemento de datos de transcripción 100A, que representa una transcripción de voz de uno o más elementos de datos de audio o habla recibidos 20.

5 De acuerdo con algunas modalidades, el sistema 100 puede incluir un módulo de extracción de características 112, adaptado para extraer una o más características de audio 112A de voz 20.

Por ejemplo, el módulo de extracción de
10 características 112 puede configurarse para extraer de voz 20 una o más características de audio 112A, tales como coeficientes cepstrales en la frecuencia-Mel (MFCC, por sus siglas en inglés). Como se conoce en la técnica, los coeficientes MFCC pueden derivarse de un cepstrum de
15 un elemento de datos de audio tal como un flujo de una señal de audio. El cepstrum puede, a su vez, computarse como una transformada de Fourier inversa (IFT, por sus siglas en inglés) de un logaritmo de un espectro de frecuencias de la señal de audio.

20 Como se muestra en la Figura 2, el módulo de extracción de características 112 puede alimentar características de audio 112A (p. ej., MFCC) en un modelo acústico 110 basado en aprendizaje automático (ML). El sistema 100 puede aplicar el modelo acústico 110 sobre
25 uno o más elementos de datos de voz 20 y/o sobre las características de audio correspondientes 112A, para

producir o predecir (como se hace referencia comúnmente en la técnica) una función de distribución de caracteres 110A. La función de distribución de caracteres 110A puede representar una distribución de probabilidad de caracteres lingüísticos (p. ej., [a-z]) sobre un alfabeto dado, basada en las características de audio de entrada 112A. En otras palabras, la función de distribución de caracteres 110A puede representar una probabilidad de aparición de uno o más caracteres lingüísticos en el uno o más elementos de datos de voz 20.

Como se conoce en la técnica, el modelo acústico 110 puede predecir la función de distribución de caracteres 110A en relación con marcos de tiempo específicos (p. ej., 25 milisegundos(ms)) del elemento de datos de voz de entrada 20. En otras palabras, para cada marco de tiempo de voz de entrada 20, el modelo acústico basado en ML 110 puede producir una función de distribución de caracteres 110A de caracteres lingüísticos (y/o porciones no habladas o de “espacios en blanco” en el habla 20), representando una probabilidad de que letras o caracteres específicos hayan sido dichos o pronunciados durante ese marco de tiempo.

De acuerdo con algunas modalidades, el modelo acústico 110 puede incluir un modelo basado en ML, tal como un modelo de red neuronal convolucional (CNN, por sus siglas en inglés), un modelo de red neuronal convolucional

profunda (DNN, por sus siglas en inglés), un modelo de red neuronal recurrente (RNN, por sus siglas en inglés), un modelo de red neuronal basado en la atención, o cualquier otro modelo ML apropiado para predecir la función de
5 distribución de caracteres 110A, como se conoce en la técnica.

De acuerdo con algunas modalidades, el sistema 100 puede incluir un módulo decodificador voraz 120, adaptado para recibir una función de distribución de
10 caracteres 110A del modelo acústico 110, perteneciente a un marco de tiempo específico de un elemento de datos de voz subyacente 20.

Como se sabe en la técnica, el decodificador voraz 120 puede configurarse para eliminar inicialmente las
15 repeticiones de caracteres y/o la aparición de porciones que no sean de voz o “espacios en blanco” dentro del marco de tiempo dado. El decodificador voraz 120 puede determinar posteriormente la combinación o secuencia más probable de caracteres restantes, dentro del marco de tiempo dado y/o
20 entre una pluralidad de marcos de tiempo consecutivos, para producir uno o más unigramas 120A o palabras, basado en la función de distribución de caracteres 110A. Por ejemplo, el decodificador voraz 120 puede producir uno o más unigramas 120A seleccionando una secuencia de caracteres que
25 corresponden a la probabilidad de aparición máxima, tal

como se refleja por la función de distribución de caracteres 110A.

Adicionalmente, el decodificador voraz 120 puede asociar o atribuir metadatos de unigrama 120A' a cada unigrama 120A producido. Por ejemplo, los metadatos de unigrama 120A' pueden incluir un marco de tiempo correspondiente a cada unigrama o palabra 120A producido, definiendo un tiempo de inicio y un tiempo de fin para la secuencia de caracteres que comprenden el unigrama 120A relevante.

En otro ejemplo, el decodificador voraz 120 puede calcular, para uno o más (p. ej., cada uno) unigramas producidos, un elemento de metadatos de unigrama 120A' que representa un nivel de confianza o puntuación de confianza (p. ej., un valor numérico en el intervalo de $[0,1]$) para la aparición del unigrama 120A en el elemento de datos de voz subyacente 20. La puntuación de confianza puede, por ejemplo, calcularse en función de las probabilidades de los caracteres incluidos en el unigrama (p. ej., tal como se refleja en la función de distribución de caracteres 110A), y puede normalizarse, por ejemplo, mediante la longitud (p. ej., número de caracteres) del unigrama relevante. El decodificador voraz 120 puede entonces asignar o atribuir la puntuación de confianza como metadatos 120A' para el unigrama 120A relevante.

De acuerdo con algunas modalidades, el sistema
100 puede aplicar continuamente el decodificador voraz
120 en una o más instancias de la función de distribución
de caracteres 110A para producir o predecir un corpus
5 inicial de unigramas 120A (y los metadatos 120A'
correspondientes). El término “continuamente” puede usarse
en este contexto para indicar que el decodificador voraz
120 puede aplicarse en una pluralidad de instancias de la
función de distribución de caracteres 110A, que se
10 origina a partir de una pluralidad respectiva de marcos
de tiempo en un único elemento de datos de voz 20.
Adicionalmente, o alternativamente, el decodificador
voraz 120 puede aplicarse a una pluralidad de instancias
de la función de distribución de caracteres 110A, que se
15 originan a partir de una pluralidad de elementos de datos
de voz 20.

Un experto en la técnica puede apreciar que un
decodificador voraz, tal como el decodificador voraz 120,
puede ser capaz de reconocer cualquier secuencia de
20 caracteres, pero puede ser propenso a emitir palabras mal
escritas (p. ej., “termómetro” en lugar de “termómetro”).
En otras palabras, el corpus inicial de unigramas 120A
puede incluir palabras mal escritas. Por el contrario, un
decodificador de haces, tal como el decodificador de haces
25 170, puede no producir o emitir una transcripción que
incluya palabras mal escritas, pero puede limitarse por un

vocabulario o modelo lingüístico 160A amplio, aunque finito.

Se hace referencia ahora a la Figura 3, que es un diagrama en bloque que ilustra el flujo de datos en el sistema 100 para el reconocimiento de voz, de acuerdo con algunas modalidades de la invención. Puede apreciarse que el sistema 100 de la Figura 3 puede ser el mismo sistema 100 ilustrado en la Figura 2, donde algunos de los módulos y elementos se han omitido por motivos de claridad.

De acuerdo con algunas modalidades, el decodificador voraz 120 puede colaborar con un módulo de filtro 121, configurado para aplicar un filtro en el corpus inicial de unigramas 120A, para filtrar o excluir unigramas (p. ej., palabras) que ya están incluidos en el modelo lingüístico 160A. Esta filtración puede producir un corpus de unigramas candidatos, denotados en la presente descripción como unigramas 121A y metadatos 121A' correspondientes.

Como se elabora en la presente descripción (p. ej., en relación con la Figura 4), el sistema 100 puede aplicar un conjunto de filtros en los unigramas candidatos 121A para producir uno o más unigramas filtrados, denotados en la presente descripción como unigramas filtrados 130A, 140A, y metadatos correspondientes (130A', 140A'), que describen los

unigramas filtrados 130A, 140A. Como se elabora en la presente descripción (p. ej., en relación con la Figura 4), el sistema 100 puede analizar el elemento de datos de voz de audio 20 y/o los unigramas filtrados 130A/140A para producir o extraer al menos un n-grama 150A, que puede incluir uno o más unigramas filtrados 130A/140A. El sistema 100 puede entonces actualizar o mejorar el modelo lingüístico 160A para incluir el al menos un n-grama 150A.

10 Como se muestra en la Figura 3, el sistema 100 puede incluir, o puede colaborar con un decodificador de haz 170. El sistema 100 puede permitir que el decodificador de haz 170 utilice el modelo lingüístico mejorado 160A para producir una transcripción 100A sin errores de un elemento de datos de voz de audio 20, basado en el modelo lingüístico mejorado 160A. En otras palabras, el sistema 100 puede aplicar el decodificador de haz 170 en al menos un nuevo elemento de datos de voz 20 para producir al menos una transcripción 100A del nuevo elemento de datos de voz, basado en el modelo lingüístico 160A actualizado.

Adicionalmente, y como también se elabora en la presente descripción, el sistema 100 puede utilizar los n-gramas extraídos 150A, con segmentos de audio correspondientes de los elementos de datos de voz 20 (p. ej., marcados por la hora de inicio y la hora de

finalización del n-grama en los elementos de datos de voz
20) como retroalimentación para el modelo acústico 110,
para afinar o volver a entrenar el modelo acústico 110.
Por lo tanto, el sistema 100 puede producir
5 automáticamente predicciones más precisas de la función de
distribución de caracteres 110A, basado en la extracción
automática de n-gramas 150A.

Las modalidades de la invención pueden incluir
una aplicación práctica de realizar una tarea de
10 reconocimiento y transcripción de voz, y pueden incluir
varias mejoras con respecto a los métodos y sistemas
actualmente disponibles para el reconocimiento de voz.

Por ejemplo, al combinar los beneficios de los
decodificadores codiciosos y los decodificadores de haz en
15 sinergia, las modalidades pueden identificar
automáticamente (p. ej., sin necesidad de supervisión
manual) nuevos unigramas o palabras que están ausentes en
el modelo lingüístico 160A, mejorar el modelo lingüístico o
vocabulario 160A para incluir estos nuevos unigramas o
20 palabras, y producir una transcripción sin errores de los
elementos de datos de voz que incluyen los unigramas recién
identificados.

Adicionalmente, las modalidades de la invención
pueden utilizar este beneficio de identificación
25 automatizada de unigramas para reentrenar o refinar
continuamente (p. ej., repetidamente a través del tiempo)

un modelo acústico subyacente tal como el modelo acústico 110. Puede apreciarse que tal modelo acústico puede ser una piedra angular de cualquier paradigma de reconocimiento de voz. Por lo tanto, el refinamiento o
5 afinación del modelo acústico puede, por ejemplo, dar como resultado la producción de transcripciones de texto 100A que pueden ser agnósticas a la localización o acento de un hablante.

La Figura 4 es un diagrama en bloque, que
10 ilustra otra vista del sistema 100 de reconocimiento de voz, de acuerdo con algunas modalidades de la invención. Puede apreciarse que el sistema 100 de la Figura 4 puede ser el mismo sistema 100 que se ilustra en la Figura 2 y/o la Figura 3.

15 Como se muestra en la Figura 4, el sistema 100 puede incluir dos módulos de extracción de unigramas complementarios, denotados como módulos de extracción de unigramas 130 y 140. Los módulos de extracción de unigramas 130/140 pueden recibir del decodificador voraz 120 un
20 corpus de unigramas candidatos 121A, que están desprovistos de unigramas ya incluidos en el modelo lingüístico o vocabulario 160A. Al menos un (p. ej., cada uno) módulo de extracción de unigramas 130/140 puede configurarse para clasificar los unigramas candidatos 121A recibidos como una
25 palabra mal escrita o una nueva palabra probablemente

correcta en el contexto apropiado del elemento de datos de voz 20.

De acuerdo con algunas modalidades, un primer módulo de extracción de unigramas (p. ej., 140) puede
5 utilizar un corpus de texto externo tal como resúmenes de Wikipedia para filtrar las palabras probablemente mal escritas, mientras que un segundo módulo de extracción de unigramas (p. ej., 130) puede utilizar un conjunto de filtros de errores ortográficos, sin necesidad de un
10 corpus de texto externo. Puede apreciarse que el sistema 100 puede usar cualquiera de estas funcionalidades por separado, dependiendo, por ejemplo, de la disponibilidad de corpus de texto externo. Adicional o alternativamente, el sistema 100 puede combinar la funcionalidad de los
15 módulos de extracción de unigramas 130 y 140 en sinergia, para producir nuevos unigramas (p. ej., palabras) probablemente correctamente escritos para mejorar el modelo lingüístico basado en n-gramas 160A.

También se hace referencia a la Figura 5, que es
20 un diagrama en bloque, que ilustra un ejemplo de un módulo de extracción de unigramas 130, que puede incluirse en un sistema 100 para el reconocimiento de voz, de acuerdo con algunas modalidades de la invención. Como se muestra en el ejemplo de la Figura 5, el módulo de extracción de
25 unigramas 130 puede incluir uno o más (p. ej., un conjunto o cascada) de varios (p. ej., cuatro) filtros diferentes,

indicados en la presente descripción como generador candidato inicial 131, filtro basado en distancia de Levenshtein 133, filtro de espacio faltante 135 y filtro de pares de candidatos 137. Puede apreciarse que el orden de filtración (p. ej., el orden del conjunto de filtros) puede cambiar entre modalidades de la presente invención. También son posibles otros tipos de filtro y combinaciones. Como se elabora en la presente descripción, el uno o más filtros (p. ej., 131, 133, 135 y/o 137) pueden filtrar un grupo o lista de unigramas candidatos 121A para producir una lista o grupo de unigramas filtrados 130A, y metadatos 130A' correspondientes.

De acuerdo con algunas modalidades, el generador de candidatos iniciales 131 puede configurarse para filtrar los unigramas candidatos 121A que (a) ya están incluidos en el modelo lingüístico o vocabulario 160A, o (b) tienen una puntuación de confianza de metadatos 121A' que está por debajo de un umbral de confianza predefinido (p. ej., 0,95).

Adicional o alternativamente, el generador de candidatos iniciales 131 puede configurarse para filtrar los unigramas 121A que están por debajo de un recuento mínimo predefinido (p. ej., 5). Puede apreciarse que, en esta etapa, el sistema 100 puede incluir típicamente varios cientos o miles de unigramas candidatos únicos. Algunos de estos unigramas candidatos únicos pueden ser

unigramas correctamente escritos (p. ej., palabras), mientras que otros pueden estar mal escritos.

Como se elabora en la presente descripción, el módulo de extracción de unigramas 130 (y/o 140) puede configurarse para calcular, para uno o más (p. ej., cada uno) unigramas candidatos 121A, una probabilidad de error ortográfico, lo que representa una probabilidad de que un unigrama candidato relevante 121A sea una versión con error ortográfico de un unigrama que ya está incluido en el modelo lingüístico 160A. El módulo de extracción de unigramas 130 (y/o 140) puede entonces filtrar los unigramas candidatos 121A que correspondan a una probabilidad de error ortográfico que exceda un umbral predefinido. En otras palabras, el módulo de extracción de unigramas 130 (y/o 140) puede filtrar continuamente (p. ej., repetidamente a lo largo del tiempo) los unigramas candidatos 121A de una manera que se eliminen más palabras incorrectas o mal escritas en relación con las palabras correctamente escritas.

20 Tabla 2

La Tabla 2 a continuación elabora ejemplos para varios tipos de unigramas mal escritos o palabras 121A, que pueden producirse por un decodificador voraz, y pueden ser abordados o modificados por el sistema 100 para el reconocimiento de voz, de acuerdo con algunas modalidades

de la invención. Los ejemplos proporcionados en la Tabla 2 representan ejemplos encontrados experimentalmente de unigramas mal escritos, frecuentes y de alta confianza, producidos por el decodificador voraz 120.

5

	Tipo	Versión con error ortográfico	Versión correcta	Regla de corrección
1.	Eliminación de caracteres	Fabric	Fabricar	Distancia de Levenshtein
2.	Eliminación de caracteres	Adelant	Adelante	Distancia de Levenshtein
3.	Inserción de caracteres	Ootro	Otro	Distancia de Levenshtein
4.	Inserción de caracteres	Recomenadar	Recomendar	Distancia de Levenshtein
5.	Sustitución de caracteres	Ancología	Oncología	Distancia de Levenshtein
6.	Sustitución de caracteres	Rguja	Aguja	Distancia de Levenshtein
7.	Sustitución de caracteres	Kartucho	Cartucho	Distancia de Levenshtein
8.	Espacio faltante	Teloagradezc o	Te lo agradezco	Regla de espacio probablemente faltante
9.	Espacio faltante	Bienvenidoad iós	Bienvenido adiós	Regla de espacio probablemente faltante
10.	Palabra dividida	Rec tángulo	Rectángulo	Regla de fusión de bigramas

Como se conoce en la técnica, la distancia de Levenshtein es una métrica que puede usarse para medir la diferencia entre dos secuencias. Por ejemplo, una distancia de Levenshtein entre dos palabras es el número
5 mínimo de ediciones de un solo carácter (p. ej., inserciones, eliminaciones o sustituciones) necesarias para cambiar una palabra por otra.

De acuerdo con algunas modalidades, el filtro basado en Levenshtein 133 puede configurarse para manejar
10 (p. ej., filtrar) unigramas candidatos mal escritos 121A, que manifiestan un cambio (p. ej., sustitución, inserción y/o eliminación) en al menos un carácter de una palabra correctamente escrita.

Perteneciente al ejemplo presentado en la Tabla
15 2, los ejemplos 1-7 representan unigramas mal escritos 121A (p. ej., “Adelant”), que tienen errores ortográficos que ocurrieron debido a la sustitución, inserción y/o eliminación de caracteres (p. ej., en relación con “Adelante”). En otras palabras, el filtro basado en
20 Levenshtein 133 puede configurarse para filtrar al menos un unigrama candidato 121A que está lo suficientemente cerca (p. ej., en términos de la distancia de Levenshtein) a las palabras en el vocabulario.

Puede apreciarse que el filtrado de distancia
25 de Levenshtein como se conoce en la técnica que puede filtrar unigramas 121A basado únicamente en la métrica de

la distancia de Levenshtein entre unigramas puede resultar demasiado agresivo, en el sentido de que puede excluir innecesariamente palabras legítimas (p. ej., correctamente escritas). Por ejemplo, en el caso de que
5 exista una pluralidad de variantes de una palabra correctamente escritas y dentro del vocabulario, un filtro basado en Levenshtein actualmente disponible haría coincidir un unigrama candidato legítimo 121A con una variante de la palabra y, por lo tanto, podría
10 descalificar y filtrar erróneamente el unigrama candidato 121A.

De acuerdo con algunas modalidades, el filtro basado en Levenshtein 133 puede implementar una o más pruebas adicionales con el fin de evitar tales errores y,
15 por lo tanto, puede proporcionar una mejora con respecto a los filtros de distancia de Levenshtein como se conoce en la técnica.

Por ejemplo, el filtro basado en Levenshtein 133 puede calcular un valor de distancia de Levenshtein 133A
20 entre el unigrama candidato 121A y al menos un unigrama en vocabulario, ya incluido en el modelo lingüístico 160A. El filtro basado en Levenshtein 133 puede entonces calcular una puntuación de frecuencia 133B, que representa una relación de apariencia entre el unigrama candidato 121A y
25 el al menos un unigrama en vocabulario en el uno o más elementos de datos de voz 20. El filtro basado en

Levenshtein 133 puede calcular posteriormente una probabilidad de error ortográfico 133E, basado en el valor de distancia de Levenshtein 133A y el valor de puntuación de frecuencia 133B.

5 Por ejemplo, el filtro basado en Levenshtein 133 puede calcular un valor de distancia de Levenshtein entre los unigramas candidatos 121A y los unigramas ya incluidos en el modelo lingüístico 160A. El filtro basado en Levenshtein 133 puede identificar uno o más de tales pares
10 como pares similares, de acuerdo con el valor métrico de distancia (p. ej., que tiene un valor de distancia de Levenshtein que está por debajo de un umbral predefinido). El filtro basado en Levenshtein 133 puede preservar los unigramas candidatos 121A que tienen un valor de puntuación
15 de frecuencia 133B más alto (p. ej., aparecen más veces en los elementos de datos de voz 20) en comparación con su unigrama de par similar en el vocabulario 160A.

 Por ejemplo, si un unigrama candidato 121A fuera de vocabulario (OOV, por sus siglas en inglés) tal como
20 “presurización” aparece 15 veces en un corpus de elementos de datos de voz 20 de entrada, y un unigrama en el vocabulario (p. ej., incluido en el modelo lingüístico 160A) como “presión” aparece dos veces en el corpus de elementos de datos de voz de entrada 20, entonces el
25 filtro basado en Levenshtein 133 puede preservar el unigrama candidato 121A “presurización”. En otro ejemplo,

si un unigrama candidato OOV 121A tal como “insecto” aparece 17 veces en un corpus de elementos de datos de voz de entrada 20, y un unigrama en el vocabulario como “inyecto” aparece 3580 veces, entonces el filtro 133 basado en Levenshtein puede filtrar “insecto” de la lista de unigramas candidatos 121A.

Como se elabora en la presente descripción (p. ej., en relación con la Figura 2) el decodificador voraz 120 puede configurarse para emitir o calcular, para cada unigrama predicho del corpus inicial de unigramas 120A, un respectivo elemento de metadatos de unigrama 120A', que representa un nivel de confianza del unigrama predicho. De acuerdo con algunas modalidades, el filtro basado en Levenshtein 133 puede calcular la probabilidad de error ortográfico 133E, basado además en el nivel de confianza (p. ej., además del valor de distancia de Levenshtein 133A y el valor de puntuación de frecuencia 133B).

Por ejemplo, el filtro basado en Levenshtein 133 puede calcular una puntuación de confianza promedio 133D, que representa un promedio del nivel de confianza 120A' para una o más apariciones del unigrama candidato 121A en el corpus de elementos de datos de voz 20. El filtro basado en Levenshtein 133 puede calcular una suma ponderada del valor de puntuación de frecuencia 133B y la puntuación de confianza promedio 133D. El filtro basado

en Levenshtein 133 puede entonces preservar los unigramas
candidatos 121A que tienen una suma ponderada más alta
del valor de puntuación de frecuencia 133B puntuación de
confianza promedio 133D en comparación con su unigrama de
5 par similar 160A en el vocabulario.

Adicional o alternativamente, el filtro basado
en Levenshtein 133 puede preservar los unigramas
candidatos 121A basado en un conjunto predefinido de
reglas específicas del lenguaje dedicadas.

10 Por ejemplo, el modelo lingüístico 160A puede
incluir una definición de una o más reglas sintácticas
específicas del lenguaje. Por ejemplo, en el caso del
idioma inglés, la una o más reglas sintácticas
específicas del lenguaje pueden definir la conjugación
15 verbal en el tiempo (p. ej., la adición de “aba” al final
de un verbo para significar tiempo pasado), conjugación
plural vs. singular (p. ej., añadir “s” para significar
una pluralidad de sustantivos), y lo similar.

En tales modalidades, el filtro basado en
20 Levenshtein 133 puede calcular un número de ediciones o
cambios de un solo carácter entre un unigrama candidato
121A y un unigrama en vocabulario, y calcular el valor de
distancia de Levenshtein 133A basado en una o más reglas
sintácticas del lenguaje y el número de ediciones de un
25 solo carácter.

Perteneciente al ejemplo de conjugación anterior, el filtro basado en Levenshtein 133 puede preservar los unigramas candidatos 121A basado en la conjugación temporal (p. ej., la forma pasada “evaluada” vs. la forma presente “evalúa”) y/o la conjugación plural vs. la singular (p. ej., “prescripción” vs. “prescripciones”), y lo similar. También pueden implementarse otras reglas de este tipo para preservar los unigramas candidatos 121A.

10 Puede apreciarse que los filtros de distancia de Levenshtein actualmente disponibles como se conoce en la técnica pueden no distinguir entre tipos de cambios de carácter.

 Por ejemplo, una “m” sustituida por una “n”
15 puede ser un error frecuente, debido al parecido fonético de ambas letras. En cambio, la sustitución de la “d” por la “o” no es tan frecuente. Por lo tanto, es más probable que la sustitución de la “m” por la “n” represente un error ortográfico que la sustitución de la “d” por la “o”.
20 En otro ejemplo, la inserción errónea de la vocal “i” en los sistemas actualmente disponibles de reconocimiento de voz puede ocurrir con mayor frecuencia que la inserción de la consonante “t” y, por lo tanto, es más probable que represente un error ortográfico.

De acuerdo con algunas modalidades, el filtro basado en Levenshtein 133 puede tener en cuenta la frecuencia de tales cambios (p. ej., sustituciones, eliminaciones y/o inserciones) y, por lo tanto, puede proporcionar una mejora con respecto a los filtros de distancia de Levenshtein como se conoce en la técnica. Por ejemplo, el módulo de extracción de unigramas puede cuantificar cambios (p. ej., sustituciones, eliminaciones y/o inserciones) de caracteres calculando las probabilidades de cambio 133C basado en la alineación a nivel de caracteres de la transcripción del decodificador voraz vs. la transcripción humana, como se explica en la presente descripción (p. ej., en relación con la Figura 6).

Se hace referencia ahora a la Figura 6, que es un ejemplo de un mapa de calor que ilustra la probabilidad de cambio 133C (p. ej., sustitución y/o eliminación) de letras del alfabeto inglés en un sistema de reconocimiento de voz, de acuerdo con algunas modalidades de la invención.

El ejemplo de mapa de calor de la Figura 6 se ha obtenido basado en elementos de datos de voz de audio que abarcan varias horas y su correspondiente transcripción. Se calcularon las probabilidades de cambio 133C para cada carácter como una relación entre la frecuencia o el número

de ocurrencias de cambio y el número total de apariciones del carácter en el corpus de elementos de datos de voz 20.

Por ejemplo, y como se muestra en el mapa de calor de la Figura 6, las letras “z” y “s” tienen una alta probabilidad de sustitución, al igual que “q” y “c”. En otro ejemplo, el carácter apóstrofe (') tiene una probabilidad de eliminación muy grande.

De acuerdo con algunas modalidades, el filtro basado en Levenshtein 133 puede calcular un valor de distancia de Levenshtein 133A entre el unigrama candidato 121A y al menos un unigrama en vocabulario, ya incluido en el modelo lingüístico 160A. Para cada carácter que contribuye al valor de distancia de Levenshtein 133A, el filtro basado en Levenshtein 133 puede calcular una probabilidad de un cambio subyacente 133C. El filtro basado en Levenshtein 133 puede calcular posteriormente una probabilidad de error ortográfico 133E, basado en el valor de distancia de Levenshtein 133A y, además, en el valor de puntuación de frecuencia 133B y/o probabilidad de cambio 133C. Por ejemplo, la probabilidad de error ortográfico 133E puede calcularse como una suma ponderada del valor de distancia de Levenshtein 133A, ponderada por la probabilidad de cambio 133C. El filtro basado en Levenshtein 133 puede entonces filtrar uno o más unigramas candidatos 121A basado en la probabilidad de error ortográfico 133E.

Como se muestra en los ejemplos de la Tabla 2 (p. ej., las entradas 8 y 9), un decodificador voraz puede producir unigramas candidatos mal escritos 121A debido a la omisión de un espacio o “espacio en blanco” entre dos unigramas consecutivos (p. ej., palabras). Por ejemplo, el decodificador voraz 120 puede omitir un espacio en blanco entre las palabras “Te”, “lo” y “agradezco”, para producir un unigrama candidato 121A mal escrito como “Teloagradezco”.

De acuerdo con algunas modalidades, el módulo de extracción de unigramas 130 puede incluir un módulo de filtro de espacio faltante 135, configurado para omitir o filtrar dichos unigramas candidatos mal escritos 121A.

En otras palabras, el módulo de filtro de espacio faltante 135 puede calcular, para uno o más unigramas candidatos 121A, una probabilidad de espacio faltante, lo que representa una probabilidad de que el unigrama candidato 121A sea una concatenación de dos unigramas que ya están incluidos en el modelo lingüístico 160A. El módulo de filtro de espacio faltante 135 puede filtrar subsecuentemente los unigramas candidatos 121A que corresponden a una probabilidad de espacio faltante 135A que excede un umbral predefinido.

Por ejemplo, el módulo de filtro de espacio faltante 135 puede calcular: (a) una frecuencia de aparición de unigramas candidatos 121A (p. ej.,

“Teloagradezco”) en un corpus de elementos de datos de voz 20, y (b) una frecuencia de aparición de bigramas concatenados (p. ej., hechos de unigramas consecutivos, como “Te”, “lo” y “agradezco”) que producen los mismos unigramas que los unigramas candidatos 121A después de la concatenación (p. ej., “Teloagradezco”). El módulo de filtro de espacio faltante 135 puede calcular una probabilidad de espacio faltante 135A, basado en (p. ej., como una relación de) estas frecuencias calculadas de apariciones, y puede filtrar los unigramas candidatos 121A que corresponden a una probabilidad de espacio faltante que excede un umbral predefinido.

Adicional o alternativamente, el filtro de espacio faltante 135 puede filtrar todos los unigramas candidatos “concatenados” 121A (p. ej., “Teloagradezco”) que son menos frecuentes que la combinación correspondiente de unigramas no concatenados (p. ej., “Te”, “lo” y “agradezco”) de la lista de unigramas candidatos 121A. En este ejemplo, la lógica detrás de tal filtrado se basa en el hecho de que “Te lo agradezco” aparecería con más frecuencia en el corpus de elementos de datos de voz 20 que “Teloagradezco” y, por lo tanto, “Teloagradezco” probablemente esté mal escrito y debería eliminarse.

De acuerdo con algunas modalidades, el módulo de extracción de unigramas 130 puede incluir un módulo de filtro de pares candidatos 137, configurado para omitir o filtrar unigramas candidatos 121A que no tienen un unigrama par en el vocabulario conocido del modelo lingüístico 160A, pero sí tienen un unigrama candidato par 121A más frecuente en la lista de unigramas candidatos 121A.

De acuerdo con algunas modalidades, el módulo de filtro de par candidato 137 puede calcular un valor métrico de la distancia (p. ej., un valor de distancia de Levenshtein, como se elabora en la presente descripción) entre uno o más (p. ej., cada uno) pares de unigramas candidatos 121A. El módulo de filtro de pares candidatos 137 puede identificar uno o más de tales pares como pares similares, de acuerdo con el valor métrico de la distancia, p. ej., que tiene un valor de distancia de Levenshtein que está por debajo de un umbral predefinido. Para cada miembro (p. ej., el unigrama candidato 121A) de cada par similar, el módulo de filtro de par candidato 137 puede calcular una frecuencia de aparición en un corpus de elementos de datos de voz 20 y posteriormente eliminar o filtrar el miembro menos frecuente.

Por ejemplo, el modelo lingüístico 160A puede estar inicialmente desprovisto del unigrama (p. ej., palabra) “diabetes”. Con el tiempo, una lista de unigramas

candidatos 121A puede evolucionar para incluir al candidato de unigrama 121A correcto “diabetes”, y el candidato de unigrama mal escrito 121A “diabete”. En este ejemplo, el módulo de filtro de pares candidatos 137 puede configurarse para filtrar u omitir el unigrama candidato mal escrito 121A “diabete” de la lista de unigramas candidatos 121A, y mantener el unigrama candidato 121A correcto “diabetes” para su análisis posterior, como se elabora en la presente descripción.

También se hace referencia a la Figura 7, que es un diagrama en bloque, que ilustra otro ejemplo de un módulo de extracción de unigrama 140, que puede incluirse en un sistema de reconocimiento de voz 100, de acuerdo con algunas modalidades de la invención.

Las flechas en la Figura 7 pueden representar el flujo de datos entre módulos del módulo de extracción de unigramas 140, y/o hacia o desde el módulo de extracción de unigramas 140. Algunas flechas se han omitido para fines de claridad.

Como se muestra en la Figura 7, el módulo de extracción de unigramas 140 puede recibir uno o más (p. ej., una lista) de los unigramas candidatos 121A del decodificador voraz 120, y puede aplicar un proceso de filtrado en la lista de unigramas candidatos 121A para obtener uno o más unigramas filtrados 140A y metadatos correspondientes.

De acuerdo con algunas modalidades, el módulo de extracción de unigramas 140 puede recibir (p. ej., desde el dispositivo de entrada 7 de la Figura 1) un corpus de documentos externo que puede incluir una pluralidad de documentos de texto 40. El término “externo” puede usarse en este contexto para indicar que los documentos 40 pueden no pertenecer al mismo dominio del tema que los elementos de datos de voz 20, y pueden no ser el resultado de la transcripción del sistema 100.

Cada documento 40 puede incluir una pluralidad de unigramas de documento (p. ej., palabras) 40A. Por ejemplo, el corpus de documentos puede incluir una pluralidad de documentos en línea 40 que son resúmenes de artículos, tales como resúmenes de Wikipedia, y los unigramas de documentos 40A pueden ser palabras incluidas en los resúmenes de Wikipedia. Como se elabora en la presente descripción, el módulo de extracción de unigramas 140 puede utilizar el corpus de documentos de texto 40 para determinar qué unigramas candidatos 121A producidos por el decodificador voraz 120 son probablemente erróneos, y filtrarlos.

De acuerdo con algunas modalidades, el módulo de extracción de unigramas 140 puede incluir un modelo de incrustación 142 tal como un modelo de incrustación word2vec 142, como se conoce en la técnica.

El modelo de incrustación 142 puede entrenarse, basado en el corpus de unigramas producidos por el decodificador voraz 120 (p. ej., unigramas 120A, 121A), para calcular o emitir vectores de incrustación 142A. En
5 otras palabras, para cada muestra de un unigrama candidato 121A, el modelo de incrustación 142 puede producir un vector de incrustación 142A que puede ser o puede incluir una representación vectorial de un significado semántico del correspondiente unigrama
10 candidato 121A.

Como se elabora en la presente descripción, el módulo de extracción de unigramas 140 puede utilizar puntuaciones de similitud vectorial incrustadas junto con la distancia de Levenshtein, las puntuaciones en el dominio
15 y las frecuencias de unigrama para reducir el ruido proveniente del decodificador voraz 120 y el corpus de documentos 40.

Una combinación de tales herramientas puede ser necesaria porque los documentos 40 (p. ej., resúmenes de
20 Wikipedia) pueden contener unigramas 40A (p. ej., palabras) que pueden ser producidos por el decodificador voraz 120, pero que sin embargo pueden ser incorrectos en el contexto de una transcripción del elemento de datos de voz 20.

Por ejemplo, un elemento de datos de voz 20
25 puede incluir la palabra “yendo”, que puede ser transcrita erróneamente por el decodificador voraz 120

como “yen”, que en sí mismo es una palabra escrita correctamente que puede aparecer en el corpus de documentos externos 40. Con este ejemplo puede quedar claro que una combinación de (a) una distancia de Levenshtein suficientemente pequeña y (b) un valor de similitud del vector de incrustación o una puntuación del dominio suficientemente grande puede indicar un posible unigrama mal escrito.

En otras palabras, y como se elabora en la presente descripción, las modalidades de la invención pueden calcular una probabilidad de error ortográfico basada en, o como una función (p. ej., una suma ponderada) de un valor de distancia de Levenshtein 133A, una puntuación de similitud incrustada 142B, una puntuación del dominio y frecuencia de aparición.

Puede apreciarse por una persona experta en la técnica que la extracción de unigramas realizada por el módulo de extracción de unigramas 140 puede ser complementaria, y puede trabajar en sinergia con la extracción de unigramas realizada por el módulo de extracción de unigramas 130.

Por ejemplo, el módulo de extracción de unigramas 130 puede no depender de un corpus suficientemente grande de documentos externos 40. Por otro lado, el módulo de extracción de unigramas 140 puede permitir reducir el nivel de confianza inicial 120A’ de los unigramas candidatos (en

relación con el usado para el módulo de extracción de unigramas 130), permitiendo así que se consideren más candidatos, ampliando el conjunto final de palabras descubiertas.

5 De acuerdo con algunas modalidades, el módulo de extracción de unigramas 140 puede incluir un módulo de preprocesamiento 141, configurado para procesar o alinear una sintaxis de unigramas de documentos 40A y/o unigramas candidatos 121A.

10 Por ejemplo, el módulo de preprocesamiento 141 puede preparar documentos externos 40 (p. ej., resúmenes de Wikipedia) para que estén en minúsculas, sin puntuación ni caracteres especiales.

En otro ejemplo, el módulo de preprocesamiento
15 141 puede encontrar colocaciones en el corpus de resultados del decodificador voraz de unigramas candidatos 121A (p. ej., “solucionar problemas”) y preparar un clon 121A’ del corpus con todos los unigramas colocados “pegados” con un guion bajo (p. ej.,
20 trouble_shoot). Esto puede hacerse analizando frecuencias de coocurrencias vs. ocurrencias únicas en el corpus de elementos de datos de voz 20 y después decidiendo, basado en un valor umbral predefinido, si se ha encontrado una colocación.

25 De acuerdo con algunas modalidades, el modelo de incrustación 142 (p. ej., modelo word2vec) puede

entrenarse para producir el vector de incrustación 142A, basado en el corpus de unigramas clonados 121A'.

De acuerdo con algunas modalidades, el sistema 100 puede diferenciar entre dos tipos de unigramas, que pueden denominarse en la presente descripción unigramas del dominio y unigramas de fondo. El término “en el dominio” puede usarse en este contexto para indicar unigramas que están dentro de un dominio de tema específico en el que el sistema 100 se configura para funcionar. El término “fondo” puede usarse en este contexto para indicar unigramas que pueden estar más allá del dominio de tema específico en el que el sistema 100 se configura para funcionar.

Por ejemplo, el sistema 100 puede desplegarse para realizar el reconocimiento de voz, p. ej., en un centro de llamadas de una empresa farmacéutica. En una aplicación de este tipo, los elementos de datos de voz pueden incluir discusiones con respecto al dominio del tema de la farmacia y la medicina. En este sentido, los unigramas del dominio pueden incluir palabras tomadas del dominio del tema de la farmacia y la medicina, tales como “insulina”, “ritmo cardíaco”, “leucocitos”, “enfisema”, etc.

Perteneciente al mismo ejemplo, los unigramas comunes como “hola”, “gracias”, “no”, “qué”, “hoy” etc. no

son únicos ni específicos de ningún dominio del tema y, por lo tanto, también pueden denominarse palabras de dominio de fondo o fuera de dominio. Puede apreciarse que la abundancia de la aparición de tales palabras a través de una pluralidad de dominios del tema es lo que hace que estas palabras sean “ruidosas”, en el sentido de que pueden añadir datos irrelevantes al procesamiento de los elementos de datos de voz 20. Adicionalmente, las palabras tomadas de un dominio del tema de la banca y seguros, como “hipoteca” y “crédito”, pueden considerarse unigramas de dominio de fondo, ya que no son unigramas de dominio con respecto al dominio del tema de la farmacia y la medicina.

De acuerdo con algunas modalidades, el módulo de extracción de unigramas 140 puede incluir un módulo generador de corpus de fondo 143 (o “módulo de fondo 143” para abreviar). El módulo de fondo 143 puede configurarse para recibir una pluralidad de corpus de texto de fondo 20C’. Por ejemplo, los corpus de texto de fondo 20C’ pueden incluir una pluralidad de elementos de datos de texto, obtenidos mediante la transcripción de uno o más (p. ej., una pluralidad de) elementos de datos de voz de fondo 20C. El término “fondo” puede usarse en este contexto para indicar que los elementos de datos de voz 20C y/o los corpus de texto subsiguientes 20C’ pueden no

relacionarse con el dominio del tema de los elementos de datos de voz 20.

De acuerdo con algunas modalidades, el módulo de fondo 143 puede filtrarse unigramas poco frecuentes de los
5 cuerpos de texto recibidos 20C' para producir un corpus de unigramas de fondo 143A. La motivación para tal filtrado puede ser la filtración (p. ej., omisión) de unigramas mal escritos en los cuerpos de texto recibidos 20C'.

De acuerdo con algunas modalidades, el módulo
10 de extracción de unigramas 140 puede incluir una calculadora de puntuación en el dominio, adaptada para calcular una puntuación del dominio 144A para cada unigrama candidato 121A en el corpus decodificado voraz 120A.

15 Por ejemplo, el módulo de extracción de unigramas 140 también puede calcular una segunda frecuencia de aparición del unigrama candidato 121A en el corpus codificador voraz de unigramas 120A. El módulo de extracción de unigramas 140 puede normalizar (p. ej.,
20 dividir) la segunda frecuencia de aparición por el número total de documentos o elementos de datos de voz 20. Esta frecuencia de aparición puede denominarse en la presente descripción como una frecuencia de documento normalizada en primer plano 144B. El módulo de extracción de
25 unigramas 140 puede calcular una primera frecuencia de aparición de los unigramas candidatos 121A en el corpus

de unigramas de fondo 143A. El módulo de extracción de unigramas 140 puede normalizar (p. ej., dividir) la primera frecuencia de aparición por el número total de documentos en los corpus de texto recibidos 20C'. Esta frecuencia de aparición puede denominarse en la presente descripción como una frecuencia de documento normalizada de fondo 144C. El módulo de extracción de unigramas 140 puede dividir posteriormente la frecuencia de documento normalizado en primer plano 144B por la frecuencia de documento normalizado de fondo 144C para obtener una puntuación del dominio 144A para cada unigrama candidato 121A.

Como se elabora en la presente descripción, el decodificador voraz 120 puede emitir, para cada unigrama decodificado 120A, un elemento de metadatos 120A' tal como metadatos de nivel de confianza 120A'. De acuerdo con algunas modalidades, el módulo de extracción de unigramas 140 puede incluir un módulo de puntuación de confianza 149, adaptado para recibir, para cada unigrama candidato 121A del corpus de unigramas decodificado voraz 120A, una puntuación de confianza 149A. La puntuación de confianza 149A puede representar, por ejemplo, una confianza promedio para cada unigrama. Esto puede calcularse, por ejemplo, promediando todas las ocurrencias de un unigrama 120A en la salida decodificada

voraz, y calculando un promedio de sus metadatos de nivel de confianza 120A'.

Como se conoce en la técnica del procesamiento del lenguaje natural, un vector de incrustación de palabras puede ser una representación vectorial de un significado semántico de una palabra, de tal manera que se espera que las palabras que están más cerca en el espacio vectorial sean similares en significado. El vector de incrustación de palabras puede obtenerse o producirse mediante un modelo de incrustación (p. ej., un modelo basado en ML), basado en un corpus predefinido de palabras o unigramas.

De acuerdo con algunas modalidades, el módulo de extracción de unigramas 140 puede producir, a partir de los unigramas candidatos 121A (o 121A') una pluralidad de combinaciones de pares 122, donde cada par 122 incluye dos unigramas candidatos 121A, y en donde cada par se asocia con uno o más valores métricos de pares 122A, definiendo la relación entre los miembros de los pares de unigramas candidatos 122.

Como se elabora en la presente descripción, el modelo de incrustación (p. ej., modelo word2vec) 142 puede producir, para al menos un primer unigrama candidato 121A (o 121A') un primer vector de incrustación de palabras 142A, basado en el corpus de unigramas candidatos 121A (o 121A'), y producir, para al menos un segundo unigrama

candidato 121A (o 121A') un segundo vector de incrustación de palabras 142A basado en el corpus de unigramas candidatos 121A (o 121A').

De acuerdo con algunas modalidades, el modelo de
5 incrustación 142 puede calcular, para uno o más (p. ej., cada uno) pares 122 de unigramas candidatos 121A (o 121A') un valor métrico de pares 122A que es una puntuación de similitud 142B basada en el primer vector de incrustación de unigramas 142A y el segundo vector de incrustación de
10 unigramas 142A. Por ejemplo, la puntuación de similitud 142B puede calcularse como una similitud de coseno entre los dos miembros del par de unigramas candidatos 121A. En algunas modalidades, el modelo de incrustación 142 solo puede producir un vector de incrustación 142A, y/o calcular
15 la puntuación de similitud 142B basado en la puntuación de confianza 149A (p. ej., para los unigramas candidatos 121A que tienen una puntuación de confianza 149A por encima de un umbral predefinido).

Adicional o alternativamente, el módulo de
20 extracción de unigramas 140 puede calcular, para uno o más (p. ej., cada uno) pares 122 de unigramas candidatos 121A (o 121A') un valor métrico de pares 122A que es un valor de distancia de Levenshtein, como se elabora en la presente descripción (p. ej., en relación con la Figura 5).

25 Adicional o alternativamente, el módulo de extracción de unigramas 140 puede calcular, para uno o

más (p. ej., cada uno) pares 122 de unigramas candidatos 121A (o 121A') un valor métrico de pares 122A que es una frecuencia de unigrama de aparición para cada unigrama miembro 121A en el par 122.

5 Adicional o alternativamente, el módulo de extracción de unigramas 140 puede calcular, para uno o más (p. ej., cada uno) pares 122 de unigramas candidatos 121A un valor métrico de pares 122A que es una puntuación del dominio 144C para cada miembro de unigrama 121A en el par
10 122. Como se elabora en la presente descripción, el módulo de extracción de unigramas 140 puede calcular la puntuación del dominio 144C basado en la pluralidad de unigramas de documentos 40A, de tal manera que la puntuación del dominio 144C puede representar una probabilidad de que el unigrama
15 candidato sea pertinente para al menos un dominio específico.

 Adicional o alternativamente, el módulo de extracción de unigramas 140 puede calcular, para uno o más (p. ej., cada uno) pares 122 de unigramas candidatos
20 121A un valor métrico de pares 122A que es una puntuación de confianza (p. ej., un nivel de confianza promedio) para cada miembro de unigrama 121A en el par 122.

 De acuerdo con algunas modalidades, el módulo de extracción de unigramas 140 puede incluir un módulo
25 generador de listas de candidatos 145, adaptado para filtrar los unigramas candidatos 121A (o 121A') basado en

valores métricos de pares 122A y/o basado en los respectivos metadatos de unigrama 120A' para producir una lista inicial de unigramas candidatos 145A. En otras palabras, la lista inicial de unigramas candidatos 145A
5 puede incluir los unigramas candidatos 121A obtenidos por el decodificador voraz 120, excepto para los unigramas candidatos 121A que fueron filtrados, como se elabora en la presente descripción.

Por ejemplo, el módulo generador de listas 145
10 puede filtrar candidatos de unigramas 121A que tienen una puntuación de confianza 149A que está por debajo de un umbral de confianza predefinido. Adicional o alternativamente, el módulo generador de listas 145 puede filtrar candidatos de unigramas 121A que tienen un
15 recuento de apariencias, o una frecuencia de aparición que está por debajo de un umbral predefinido en los elementos de datos de voz 20. Adicional o alternativamente, el módulo generador de listas 145 puede filtrar candidatos de unigramas 121A que no aparecen en el corpus de
20 unigramas de documentos 40A. Adicional o alternativamente, el módulo generador de listas 145 puede filtrar candidatos de unigramas 121A que están en el vocabulario (p. ej., incluidos en el vocabulario del modelo lingüístico 160A).

25 De acuerdo con algunas modalidades, el módulo de extracción de unigramas 140 puede incluir uno o más filtros

de erróneos ortográficos 146, configurados para filtrar candidatos de unigramas 121A de la lista de unigramas candidatos 145A, como se elabora en la presente descripción.

5 Un primer filtro de error ortográfico 146 puede ser un filtro de error ortográfico 146 de palabras comunes, en el sentido de que puede configurarse para filtrar los unigramas candidatos 121A de la lista de unigramas candidatos 145A como probables errores ortográficos de
10 palabras comunes (p. ej., palabras sin dominio, que tienen una puntuación del dominio 144C que está por debajo de un umbral predefinido).

 Como se desarrolla en la presente descripción, las palabras comunes (p. ej., no específicas de dominio)
15 tales como: “es”, “mí”, “yo”, “tener”, “por qué” etc. pueden tener la misma frecuencia en todos los dominios de los centros de llamadas (ya sea de moda, telecomunicaciones, farmacia, etc.), ya que las modalidades elaboradas pueden normalizar la puntuación del dominio 144C dividiendo la
20 frecuencia de aparición de una palabra relevante en el corpus del dominio por la frecuencia de aparición en un corpus de fondo. Por lo tanto, la puntuación del dominio 144C de palabras comunes puede tener un valor cercano a 1,0.

25 De acuerdo con algunas modalidades, el filtro de error ortográfico 146 de palabras comunes puede eliminar

o filtrar un primer candidato de unigrama 121A que tiene al menos un segundo unigrama emparejado (p. ej., en pares 122), en donde el segundo unigrama candidato 121A: (a) tiene una puntuación del dominio 144C que está dentro de un valor de rango predefinido (p. ej., se sospecha que es una palabra común), o tiene una puntuación de confianza (p. ej., un nivel de confianza promedio) que está por encima de un umbral predefinido, y (b) tiene un valor de distancia de Levenshtein en relación con el primer unigrama candidato 121A que está por debajo de un umbral predefinido.

En otro ejemplo, el filtro de error ortográfico 146 de palabras comunes puede eliminar o filtrar un primer candidato de unigrama 121A que tiene al menos un segundo unigrama emparejado (p. ej., en pares 122), en donde el segundo unigrama candidato 121A tiene una frecuencia de aparición más alta en los elementos de datos la voz 20 que el primer candidato de unigrama 121A.

Además, el filtro de error ortográfico 146 de palabras comunes puede conservar una o más listas de unigramas candidatos 145A del candidato de unigrama 121A basado en reglas específicas del lenguaje. Por ejemplo, el filtro de error ortográfico 146 de palabras comunes puede conservar los unigramas candidatos del candidato de unigrama 121A que incluyen una conjugación plural (p. ej., concierto vs. conciertos).

Un segundo filtro de error ortográfico 146 puede ser un filtro de error ortográfico de palabras del dominio 146, en el sentido de que puede configurarse para filtrar los unigramas candidatos 121A de la lista de unigramas candidatos 145A como probables errores ortográficos de palabras del dominio (p. ej., unigramas que tienen una puntuación del dominio 144C que está por encima de un umbral predefinido).

Por ejemplo, el filtro de error ortográfico de palabras en el dominio 146 puede eliminar o filtrar un primer candidato de unigrama 121A que tiene al menos un segundo unigrama emparejado (p. ej., en pares 122), en donde: (a) el valor de distancia de Levenshtein entre los miembros unigramas candidatos 121A del par está por debajo de un umbral predefinido; (b) el segundo unigrama candidato 121A tiene una puntuación de confianza (p. ej., un nivel de confianza promedio) que está por encima de un umbral predefinido; y (c) el segundo unigrama candidato 121A tiene una mayor frecuencia de aparición (p. ej., por un factor predefinido) en los elementos de datos de voz que el primer candidato de unigrama 121A.

De acuerdo con algunas modalidades, el filtro de error ortográfico 146 puede calcular una probabilidad de error ortográfico 146A para uno o más unigramas candidatos 121A basado en uno o más valores métricos de pares 122A. Por ejemplo, el filtro de error ortográfico

146 puede calcular la probabilidad de error ortográfico 146A como una función (p. ej., una suma ponderada) de un valor de distancia de Levenshtein 133A, una puntuación de similitud incrustada 142B, una puntuación del dominio 5 144C y/o una frecuencia de aparición 144B. El filtro de error ortográfico 146 puede filtrar subsecuentemente los unigramas candidatos 121A para los cuales la probabilidad de error ortográfico 146A excede un umbral predefinido.

De acuerdo con algunas modalidades, el módulo de 10 extracción de unigramas 140 puede incluir un módulo generador de vectores de contexto 147, configurado para producir, para cada primer unigrama candidato 121A de los unigramas candidatos 121A restantes (p. ej., después de la filtración del generador de listas de candidatos 145 y 15 filtros de errores ortográficos 146) un vector de contexto 147A.

Por ejemplo, para cada unigrama candidato restante específico 121A (p. ej., restante después de las etapas previas del generador de listas 145 y los filtros 20 de errores ortográficos 146), el módulo generador de vectores de contexto 147 puede preparar una primera lista provisional de unigramas, tomada de la unión de los unigramas transcritos del decodificador voraz 121A, que se origina a partir de elementos de datos de voz 20 que 25 incluyen unigramas candidatos 121A y palabras en el vocabulario (p. ej., unigramas ya existentes en el modelo

lingüístico 160A). El módulo generador de vectores de contexto 147 puede entonces eliminar de la primera lista provisional todos los unigramas que tienen una puntuación del dominio 144C que está por debajo de un umbral predefinido para producir una segunda lista provisional. El módulo generador de vectores de contexto 147 puede preparar una tercera lista provisional de todos los unigramas que aparecen en los mismos elementos de datos de voz 20 que el unigrama candidato específico 121A. El módulo generador de vectores de contexto 147 puede intersecar posteriormente la segunda lista provisional con la tercera lista provisional, descartando duplicados. El resultado de esta intersección puede denominarse en la presente descripción un vector de contexto 147A, perteneciente al unigrama candidato específico 121A.

Puede apreciarse que el vector de contexto 147A puede ser una lista que incluye una pluralidad de unigramas que pertenecen al mismo contexto del unigrama candidato específico 121A.

Por ejemplo, una aplicación experimental del sistema 100 ha demostrado que el unigrama candidato 121A “rodeo” puede tener un vector de contexto 147A que puede incluir unigramas tales como: “chicago”, “club”, “concierto”, “vaquero”, “juego”, “lugar”, “presentar”, “jardín” y lo similar. Cada uno de los unigramas del vector de contexto 147A tiene una puntuación del dominio

que está por encima del umbral predefinido mencionado anteriormente.

En otro ejemplo experimental del dominio de la materia de deportes, el unigrama candidato 121A “dodgers”
5 ha producido un vector de contexto 147A que incluye: “ángeles”, “béisbol”, “boston”, “chicago”, “cubs”, “juego”, “fama”, “reds”, “temporada”, “estadio”, “yankees”, y lo similar.

En otro ejemplo experimental del dominio de la
10 materia de la farmacéutica, el unigrama candidato 121A “inyección” ha producido un vector de contexto 147A que incluye: “absorbido”, “administrar”, “sangre”, “dosis”, “medicamento”, “aguja”, “punción”, “medicina”, “sustancia”, “jeringas”, “localizada”, y lo similar.

15 De acuerdo con algunas modalidades, el módulo de extracción de unigramas 140 puede incluir un módulo de cálculo de puntuación de corrección 148. El módulo de cálculo de puntuación de corrección 148 puede adaptarse para calcular, para cada unigrama candidato 121A de los
20 unigramas candidatos 121A restantes (p. ej., después de la filtración del generador de listas de candidatos 145 y filtros de errores ortográficos 146), una puntuación de corrección 148A, basada en el vector de contexto 147A respectivo del unigrama candidato.

Por ejemplo, para cada unigrama candidato específico 121A, el módulo de cálculo de puntuación de corrección 148 puede iterar sobre todos los documentos (p. ej., resúmenes de Wikipedia) en el corpus de documentos 40 que contienen el unigrama candidato específico 121A.

En cada iteración, el módulo de cálculo de puntuación de corrección 148 puede intersecar el conjunto de unigramas únicos 40A del documento relevante 40 con el vector de contexto 147A del unigrama candidato 121A. Como se explica en la presente descripción, el vector de contexto o la lista de contexto 147A puede incluir un subconjunto de unigramas de documentos 40A que (a) tienen una puntuación del dominio 144C que excede un umbral predefinido y (b) aparecen en el uno o más elementos de datos de voz 20A (p. ej., de donde se obtuvo el unigrama candidato específico 121A).

En otras palabras, para cada documento 40, el módulo de cálculo de puntuación de corrección 148 puede obtener un grupo de intersección que puede incluir unigramas que aparecen en el documento y en la lista de contexto o vector de contexto 147A. El módulo de cálculo de puntuación de corrección 148 puede calcular posteriormente, para cada documento, una puntuación de corrección que representa la relevancia del unigrama candidato para el documento, basado en las puntuaciones del dominio 144C de los unigramas del documento en el grupo de intersección.

Por ejemplo, el módulo de cálculo de puntuación de corrección 148 puede calcular una puntuación 148A de corrección sumando las puntuaciones del dominio 144C de los unigramas intersecados. Además, el módulo de cálculo de puntuación de corrección 148 puede normalizar la puntuación de corrección calculada 148A, p. ej., multiplicando la puntuación de corrección 148A por una raíz cuadrada del número de unigramas (p. ej., 40A) en la intersección. A lo largo de las iteraciones, el módulo de cálculo de puntuación de corrección 148 puede mantener el valor de puntuación de corrección 148A más alto o máximo (p. ej., máximo entre todos los documentos 40) y una identificación del documento correspondiente 40.

De acuerdo con algunas modalidades, el módulo de cálculo de puntuación de corrección 148 puede proceder a encontrar la puntuación de corrección máxima 148A y el documento correspondiente 40 para uno o más (p. ej., cada uno) de los unigramas candidatos restantes 121A. El módulo de cálculo de puntuación de corrección 148 puede entonces filtrar (p. ej., excluir) unigramas candidatos 121A que corresponden a una puntuación de corrección máxima que está por debajo de un umbral predefinido.

Por ejemplo, el módulo de cálculo de puntuación de corrección 148 puede ordenar los unigramas candidatos 121A basado en sus puntuaciones de corrección 148A en orden descendente y eliminar la cola por debajo de un cierto

umbral. Puede apreciarse que esta lista ordenada de unigramas candidatos, cada uno de los cuales aparece en el documento de mejor coincidencia de contexto 40, puede proporcionar una indicación con respecto a la corrección de las palabras.

Por ejemplo, los resultados experimentales de la evaluación del unigrama candidato 121A en el dominio de la materia de la biofarmacéutica ha producido la siguiente lista del unigrama candidato 121A, con sus correspondientes puntuaciones de corrección 148A: “prescrito” - 19628, “ingrediente” - 15540, “dosis” - 15042, “próstata” - 10541, “jeringa” - 8681, “antibióticos” - 8284, “ansiedad” - 8477, “riñones” - 8046, y lo similar.

En otro ejemplo, los resultados experimentales de la evaluación del unigrama candidato 121A en el dominio de la materia de la reventa de boletos (p. ej., boletos para eventos deportivos, teatro, ópera, etc.) han producido la siguiente lista del unigramas candidatos 121A, con sus correspondientes puntuaciones de corrección 148A: “desempeño” - 5439, “yankees” - 4471, “cubs” - 3803, “ángeles” - 3803, “estadio” - 3803, “reds” - 3481, “rodeo” - 3269, “dodgers” - 2574, “braves” - 2511, “conciertos” - 2412, “eliminatória” - 2182, “nacionales” - 1869, y lo similar.

El módulo de extracción de unigramas 140 puede seleccionar o identificar los unigramas candidatos de puntuación máxima 121A (p. ej., que tienen la puntuación de corrección más alta 148A). Estos unigramas se denominan en la presente descripción unigramas identificados 140A. Los unigramas identificados 140A (como los unigramas identificados 130A) pueden entonces analizarse adicionalmente para mejorar el modelo lingüístico 160A, como se elabora en la presente descripción.

Ahora se hace referencia de nuevo a la Figura 4. De acuerdo con algunas modalidades, y tal como se elabora en la presente descripción (p. ej., en relación con la Figura 3 y/o la Figura 4), el sistema 100 puede incluir un módulo de expansión de n-gramas 150. El módulo de expansión de n-gramas 150 puede configurarse para analizar los elementos de datos de voz 20 en vista de los unigramas identificados 130A y/o 140A para extraer de los elementos de datos de voz 20 al menos un n-grama 150A. El al menos un n-grama 150A puede incluir uno o más unigramas identificados 130A/140A.

Como se elabora en la presente descripción, el decodificador voraz 120 puede calcular, para uno o más (p. ej., cada uno) unigramas producidos, un elemento de metadatos de unigramas 120A' que representa un nivel de confianza o puntuación de confianza.

De acuerdo con algunas modalidades, el módulo de expansión de n-gramas 150 puede analizar las ocurrencias de unigramas 120A en la transcripción del decodificador voraz. Para uno o más unigramas candidatos identificados 5 130A/140A, el módulo de expansión de n-gramas 150 puede ubicar en el elemento de datos de voz 20, un n-grama de unigramas adyacentes, que incluye el unigrama candidato 130A/140A.

De acuerdo con algunas modalidades, para uno o 10 más unigramas candidatos 130A/140A, el módulo de expansión de n-gramas 150 puede localizar en el elemento de datos de voz 20 un n-grama de unigramas adyacentes, que incluye el unigrama candidato 130A/140A. En otras palabras, el módulo de expansión de n-gramas 150 puede buscar a través de las 15 transcripciones del codificador voraz 120, del elemento de datos de voz 20 (que son secuencias de unigramas decodificados voraces 120A), para encontrar unigramas identificados 130A y/o 140A en las secuencias de unigramas decodificados 120A.

20 El módulo de expansión de N-gramas 150 puede usar los unigramas identificados 130A y/o 140A como unigramas “ancla” en un algoritmo de búsqueda de expansión, como se elabora en la presente descripción. El término “ancla” puede usarse en este contexto para 25 indicar un unigrama identificado (p. ej., 130A y/o 140A)

que puede marcar el comienzo del proceso de búsqueda de expansión.

El módulo de expansión de N-gramas 150 puede configurarse para encontrar secuencias de unigramas 120A (p. ej., palabras) que incluyen los unigramas identificados 130A y/o 140A, en elementos de datos de voz 20. Estas secuencias de unigramas 120A se denominan en la presente descripción n-gramas extraídos 150A. El módulo de expansión de N-gramas 150 puede actualizar posteriormente el modelo lingüístico 160A para incluir el al menos un n-grama 150A extraído.

De acuerdo con algunas modalidades, y como lo indica su nombre, el módulo de expansión de N-gramas 150 puede realizar un proceso de búsqueda de expansión en las secuencias de unigramas 120A obtenidas por el codificador voraz 120, partiendo de un unigrama ancla 130/140. El término “expansión” puede usarse en la presente descripción para indicar que se puede componer un n-grama expandiendo una secuencia de unigramas 120A en al menos una dirección desde el unigrama ancla 130/140, hasta que se cumpla una condición de parada.

Por ejemplo, partiendo de un unigrama identificado como ancla 130A/140A en una secuencia de unigramas 120A obtenida del decodificador codicioso 120, la búsqueda de expansión puede proceder al siguiente unigrama (p. ej., palabra) en ambas direcciones (p. ej., a

un unigrama anterior en las secuencias de unigramas 120A y a una palabra posterior en las secuencias de unigramas 120A).

Si (a) el siguiente unigrama 120A tiene metadatos de valor de confianza 120A' que está por encima de un umbral predefinido, y (b) el siguiente unigrama 120A también se incluye en el grupo de unigramas identificados 130A/140A y/o se incluye en el modelo lingüístico 160A, después el siguiente unigrama 120A se añade o fusiona en el n-grama en expansión 150A. Si el siguiente unigrama 120A no cumple las condiciones (a) y (b) anteriores, entonces la condición de parada se cumple en la dirección relevante y el proceso de búsqueda de expansión se detiene en esa dirección. Si la condición de parada se cumple en ambas direcciones, entonces el proceso de búsqueda de expansión finaliza para el unigrama identificado como ancla relevante 130A/140A, y se detiene la expansión del n-grama 150A. El módulo de expansión de N-gramas 150 puede entonces proceder al siguiente unigrama identificado como ancla 130A/140A en la secuencia de unigramas transcritos decodificados voraces 120A.

Puede apreciarse que el proceso de búsqueda de expansión elaborado anteriormente puede garantizar que cada unigrama o palabra en n-grama 150A pueda tener un nivel de confianza mayor 120A' que el umbral predefinido y, por lo tanto, es probable que esté escrito

correctamente. Esto actúa como un filtro adicional que aumenta la probabilidad de que el n-grama 150A sea correcto.

Por lo tanto, y de acuerdo con algunas
5 modalidades, si (a) los unigramas 120A del n-grama en expansión 150A corresponden a un nivel de confianza 120A' que excede un valor umbral predefinido, y (b) el n-grama en expansión 150A incluye al menos (p. ej., más de) un número umbral predefinido de unigramas, entonces el generador de
10 modelos lingüísticos 160 puede actualizar el modelo lingüístico 160A para incluir el al menos un n-grama extraído 150A. De cualquier otra manera, el unigrama candidato 130A/140A puede filtrarse fuera del corpus de unigramas candidatos y no puede introducirse en el modelo
15 lingüístico 160A.

De acuerdo con algunas modalidades, el módulo de expansión de N-gramas 150 puede limitar la longitud de los n-gramas 150A a un valor máximo o longitud de palabras. Por ejemplo, el módulo de expansión de N-gramas
20 150 puede limitar la longitud de n-gramas 150A a 6 palabras en cada dirección en relación con el unigrama ancla, dando como resultado una longitud máxima de n-gramas de 13 palabras (incluyendo el unigrama ancla).

Adicional o alternativamente, el módulo de
25 expansión de n-gramas 150 puede limitar la longitud de los n-gramas 150A a un valor mínimo, p. ej., a una

longitud de 3 palabras. Tal limitación puede garantizar que los n-gramas 150A puedan mantener relaciones contextuales entre los unigramas miembros en el n-grama.

De acuerdo con algunas modalidades, el módulo
5 de expansión de N-gramas 150 puede limitar la cantidad de n-gramas literalmente idénticos. Esto aporta n-gramas completos al modelo lingüístico que contiene los unigramas fuera de vocabulario extraídos.

De acuerdo con algunas modalidades, además del
10 método de extracción de n-gramas 150A “tal cual” de la secuencia de unigramas decodificados voraces 120A, el módulo de expansión de N-gramas 150 puede recuperar más n-gramas corrigiendo ocurrencias de anclaje de unigramas probablemente mal escritos 130A/140A.

15 Por ejemplo, el módulo de expansión de n-gramas 150 puede colaborar con el (los) filtro(s) de error ortográfico 146 de la Figura 7, y utilizar una tabla de similitud tal como la Tabla 2 para preparar una lista de unigramas 151A. La lista de unigramas 151A puede incluir
20 unigramas que se han escrito erróneamente, dando como resultado anclajes de unigramas 130A/140A.

Los unigramas 151A pueden, por ejemplo, incluir unigramas que son similares a al menos un unigrama ancla 130A/140A, en el sentido de que su puntuación de similitud
25 de incrustación mutua 142B (p. ej., puntuación de similitud word2vec) está por encima de un umbral predefinido (p. ej.,

0,7), y su valor de distancia de Levenshtein mutuo 133A está por debajo de un umbral predefinido (p. ej., 2). Para una o más (p. ej., cada) ocurrencias de unigrama ancla 130A/140A, el módulo de expansión de n-gramas 150 puede
5 mantener un mapeo 151B con el unigrama original, escrito correctamente 151A. Una vez que se extrae el n-grama 150A, el módulo de expansión de n-gramas 150 puede sustituir la ocurrencia del unigrama ancla 130A/140A por la versión de unigrama correctamente escrito 151A, usando mapeo 151B. En
10 el caso de que un unigrama ancla probablemente mal escrito 130A/140A corresponda a una pluralidad de la versión de unigrama correctamente escrito 151A, el módulo de expansión de n-gramas 150 puede seleccionar el correspondiente a la puntuación de similitud de incrustación mutua más alta
15 142B. Por ejemplo, “ankees” es un error ortográfico probable de “yankees”, ya que tienen una puntuación de similitud de incrustación 142B (p. ej., puntuación de similitud word2vec) de 0,853 y un valor de distancia de Levenshtein de 1.

20 Cabe señalar que puede haber casos de división de unigrama ancla 130/140, que pueden dar lugar a una extracción errónea de n-grama 150A, si no se corrige adecuadamente. Por ejemplo, un unigrama como “solucionar problemas” puede dividirse en “solucionar” y “problemas”.
25 Aunque “problemas” puede servir como un ancla de unigrama

legítimo (p. ej., correctamente escrito) por sí solo, que puede verse en otros contextos (p. ej., no necesariamente seguido de “solucionar”), aún podría ser un ancla incorrecto.

5 De acuerdo con algunas modalidades, para corregir tales casos de unigramas divididos, el módulo de expansión de n-gramas 150 puede posprocesar los n-gramas extraídos 151A. En este posprocesamiento, el módulo de expansión de n-gramas 150 puede fusionar uno o más (p.
10 ej., cada uno) bigramas 151C (p. ej., unigramas consecutivos) en n-gramas 150A, y analizar cada bigrama fusionado 151C por separado. Por ejemplo, un bigrama puede incluir los unigramas consecutivos “rec” y “tángulo”. El subsiguiente bigrama fusionado 151C puede
15 ser “rectángulo”. Si el bigrama fusionado 151C forma un unigrama existente (p. ej., ya incluido en el modelo lingüístico 160A), legítimo (p. ej., escrito correctamente), entonces el n-grama 150A puede cambiarse para que contenga el bigrama fusionado 151C en el lugar
20 del par de unigramas prefusionados.

Cabe señalar que puede haber casos en los que la fusión de un bigrama (p. ej., dos unigramas consecutivos) pueda formar un ancla de unigrama de error ortográfico. En tales condiciones, el módulo de expansión
25 de n-gramas 150 puede colaborar con el (los) filtro(s) de error ortográfico 146 de la Figura 7, y añadir una etapa

al proceso de fusión de bigramas para comprobar si el bigrama fusionado 151C forma un ancla de error ortográfico 130A/140A. De acuerdo con algunas modalidades, aquí es donde puede utilizarse la adición de guiones bajos de colocación como se elabora en la presente descripción (p. ej., en relación con el módulo de preprocesamiento 141 de la Figura 7). Los guiones bajos de la ubicación pueden identificarse e insertarse en la tabla de similitud (p. ej., la Tabla 2). El (los) filtro(s) de error ortográfico 146 puede(n) entonces eliminar el símbolo de guion bajo y puede(n) insertar el resultado como un unigrama opcional para la sustitución 151A.

Por ejemplo, los unigramas candidatos 121A “rek” y “tángulo” pueden identificarse mediante el módulo de preprocesamiento 141 como una colación, y pueden fusionarse para obtener un unigrama candidato unificado “rek_tángulo” 121A. El (los) filtro(s) de error ortográfico 146 puede(n) insertar el unigrama candidato “rek_tángulo” 121A en una tabla de similitud (p. ej., Tabla 2) con el unigrama similar “rectángulo” que después puede ser descubierto como un unigrama ancla 130A/140A, de acuerdo con el siguiente proceso: (a) El (los) filtro(s) de error ortográfico 146 puede(n) eliminar el guion bajo en “reck_tangle” para producir “rectángulo”;

(b) El (los) filtro(s) de error ortográfico 146 puede(n) mapear 151B “rektángulo” asociado con un unigrama escrito correctamente 151A “rectángulo”; (c) El módulo de expansión de n-gramas 150 podrá posprocesar un ngrama que
5 contenga “rek tángulo”, y podrá fusionar los dos componentes para formar un bigrama fusionado 151C “rektángulo”; (d) Como “rektángulo” se asigna 151B a la forma de unigrama correctamente escrito 151A “rectángulo”, el módulo de expansión de n-gramas 150
10 puede sustituir “rektángulo” por “rectángulo”, dando como resultado un n-grama correctamente escrito 150A.

De acuerdo con algunas modalidades, el módulo de expansión de n-gramas 150 puede añadir los n-gramas recién extraídos 150A al corpus de texto interno
15 existente de transcripciones de audio 20B. Por ejemplo, las transcripciones de audio internas 20B pueden incluir transcripciones de un centro de llamadas, dedicadas a debates sobre un dominio temático específico (p. ej., banca, salud, deportes, etc.) y ahora pueden actualizarse
20 con los n-gramas recién extraídos 150A, que incluyen palabras recién identificadas y correctamente escritas.

Adicionalmente, el módulo de expansión de n-gramas 150 puede añadir los n-gramas recién extraídos 150A a las transcripciones de audio 20B con algún valor
25 ponderado 20B' que puede representar una prioridad

predefinida. Por ejemplo, el peso 20B' puede significar una prioridad de fecha (p. ej., las últimas discusiones ponderadas más relevantes que las discusiones anteriores). En otro ejemplo, el peso 20B' puede
5 significar una prioridad del tema (p. ej., cuestiones de emergencia ponderadas más relevantes que las cuestiones que no son de emergencia). También pueden implementarse otros valores ponderados 20B'.

De acuerdo con algunas modalidades, el módulo
10 de expansión de n-gramas 150 puede colaborar con un modelo de generador de modelos lingüísticos de n-gramas 160 (o "generador 160", para abreviar). El generador 160 puede recibir los n-gramas recién extraídos 150A del módulo de expansión de n-gramas 150, y puede reconstruir
15 o recompilar el modelo lingüístico 160A para incluir los n-gramas añadidos recién extraídos 150A. El sistema 100 puede proceder entonces a emplear el decodificador de haz 170 con el nuevo modelo lingüístico 160A, como se elabora en la presente descripción (p. ej., en relación con la
20 Figura 2) para producir una transcripción 100A de los elementos de datos de voz entrantes 20A.

De acuerdo con algunas modalidades, el sistema 100 puede utilizar los n-gramas extraídos 150A como datos de supervisión, para volver a entrenar automáticamente, o
25 ajustar con precisión el entrenamiento del modelo acústico 110, basado en los elementos de datos de voz 20. En otras

palabras, el sistema 100 ajustar con precisión automáticamente la red neuronal 110 usando un conjunto de datos de entrenamiento que incluye enunciados de palabras recién descubiertas para facilitar el reconocimiento futuro de estas palabras en los elementos de datos de voz 20.

De acuerdo con algunas modalidades, el decodificador voraz 120 puede producir, para cada unigrama decodificado 120A, un elemento de metadatos que es una marca de tiempo del enunciado del unigrama 120A en el elemento de datos de voz 20. El módulo de expansión de n-gramas 150 puede producir, para uno o más n-gramas extraídos 150A, al menos una marca de tiempo del fragmento correspondiente 150B, basado en los metadatos de marca de tiempo 120A'. Por ejemplo, la marca de tiempo del fragmento 150B puede incluir una marca de tiempo de un comienzo de n-grama 150A (p. ej., la marca de tiempo 120A' del primer unigrama 120A en el n-grama 150A) en el elemento de datos de voz 20. En otro ejemplo, la marca de tiempo del fragmento 150B puede incluir una marca de tiempo de una terminación de n-grama 150A (p. ej., la marca de tiempo 120A' del final del último unigrama 120A en el n-grama 150A) en el elemento de datos de voz 20.

El módulo de expansión de N-gramas 150 puede colaborar con el módulo de refinamiento 180 de red neuronal artificial (ANN). El módulo de refinamiento de ANN 180 puede recibir uno o más n-gramas extraídos 150A y

las marcas de tiempo de fragmentos correspondientes 150B (p. ej., tiempos de inicio y finalización), y puede filtrar los n-gramas extraídos 150A para producir un grupo de n-gramas 150A que son más largos que un período de tiempo predefinido (p. ej., 2,5 segundos de longitud).

El módulo de refinamiento de ANN 180 puede incluir los n-gramas extraídos 150A (que ahora son transcripciones de fragmentos de los elementos de datos de voz 20 del audio) en el corpus interno de las transcripciones de audio 20B. El módulo de refinamiento de ANN 180 puede entonces volver a entrenar el modelo acústico 110 basado en el fragmento de los elementos de datos de voz 20 de audio (como se define por las marcas de tiempo 150B) y usando los n-gramas extraídos añadidos 150A como datos de supervisión.

Adicionalmente, el módulo de refinamiento de ANN 180 puede atribuir un peso a los n-gramas extraídos añadidos 150A para aumentar el reentrenamiento del modelo acústico 110 basado en estos nuevos datos de supervisión. El módulo de refinamiento de ANN 180 puede afinar el entrenamiento del modelo acústico 110 con este conjunto de datos mixtos para varias épocas (p. ej., 10). Se ha mostrado experimentalmente que tal entrenamiento puede aumentar considerablemente la recuperación de palabras en n-gramas extraídos 150A.

Además, se ha mostrado experimentalmente que la ganancia general en la tasa de errores de palabras también es considerable cuando se detiene en el mejor de los casos la mejora en el grupo objetivo establecido.

5 Sigue siendo considerable a la hora de tomar el último modelo (pero menos). Durante las pruebas, la tasa de errores de palabras de un subconjunto de enunciados (p. ej., enunciados que contienen al menos uno de los unigramas ancla) se reduce a casi el mismo valor que la

10 tasa general de errores de palabras, mientras que anteriormente era considerablemente mayor. Esto sugiere que las palabras fuera del vocabulario son inicialmente “más difíciles” de reconocer correctamente para la red neuronal, pero cuando se introducen en la red neuronal se

15 vuelven “familiares” como el resto del vocabulario conocido.

Ahora se hace referencia a la Figura 8, que es un diagrama de flujo, que ilustra un método para descubrir automáticamente unigramas en un elemento de

20 datos de voz mediante al menos un procesador de un sistema para reconocimiento de voz, de acuerdo con algunas modalidades de la invención.

Como se muestra en la etapa S1005, el al menos un procesador (p. ej., el elemento 2 de la Figura 1) puede

25 recibir un modelo lingüístico 160A, que puede incluir una

pluralidad de n-gramas. Cada uno de tales n-gramas puede incluir uno o más unigramas.

Como se muestra en la etapa S1010, el al menos un procesador 2 puede aplicar un modelo acústico de aprendizaje automático (p. ej., el modelo acústico 110 de la Figura 2) en uno o más primeros elementos de datos de voz (p. ej., el elemento de voz 20 de la Figura 4) para obtener una función de distribución de caracteres (p. ej., el elemento de distribución de caracteres 110A de la Figura 2).

Como se muestra en la etapa S1015 y S1020, el al menos un procesador 2 puede aplicar un decodificador voraz (p. ej., el decodificador voraz 120 de la Figura 4) en la función de distribución de caracteres 110A para predecir un corpus inicial de unigramas (p. ej., elementos 120A de la Figura 3). El al menos un procesador 2 puede filtrar uno o más unigramas del corpus inicial 120A para obtener un corpus de unigramas candidatos (p. ej., elementos 121A, 130A, 140A de la Figura 3) que no están incluidos en el modelo lingüístico 160A.

Como se muestra en la etapa S1025, y como se elabora en la presente descripción (p. ej., en relación con las Figuras 3-7), el al menos un procesador 2 puede analizar uno o más primeros elementos de datos de voz 20 para extraer al menos un n-grama (p. ej., el elemento

150A de la Figura 4) que incluye un unigrama candidato (p. ej., 121A, 130A, 140A).

Como se muestra en las etapas S1030 y 1035, el al menos un procesador 2 puede actualizar el modelo lingüístico 160A para incluir el al menos un n-grama extraído 150A. El al menos un procesador 2 puede posteriormente (p. ej., en una etapa de inferencia) aplicar un decodificador de haz (p. ej., el decodificador de haz 170 de la Figura 2) en al menos un segundo elemento de datos de voz 20 para producir al menos una transcripción correspondiente del segundo elemento de datos de voz 20, basado en el modelo lingüístico 160A actualizado.

A menos que se indique explícitamente, las modalidades del método descritas en la presente descripción no se limitan a un orden o secuencia particular. Además, todas las fórmulas descritas en la presente descripción están previstas solamente como ejemplos y pueden usarse otras fórmulas o fórmulas diferentes. Además, algunas de las modalidades del método o elementos de estas descritos pueden ocurrir o realizarse en el mismo punto en el tiempo.

Si bien se han ilustrado y descrito en la presente descripción determinadas características de la invención, muchas modificaciones, sustituciones, cambios y equivalentes pueden ocurrir a los expertos en la técnica.

Por lo tanto, debe entenderse que las reivindicaciones adjuntas están destinadas a cubrir todas esas modificaciones y cambios que están dentro del verdadero espíritu de la invención.

5 Se han presentado varias modalidades. Cada una de estas modalidades puede incluir, por supuesto, características de otras modalidades presentadas, y las modalidades no descritas específicamente pueden incluir diversas características descritas en la presente
10 descripción.