Un vocabulario básico de los modelos grandes de lenguaje: una explicación y 25 términos

Núria Bel
Universitat Pompeu Fabra
ORCID: 0000-0001-9346-7803
nuria.bel@upf.edu

Núria Bel és catedràtica de Tecnologies del Llenguatge al Departament de Traducció i Ciències del Llenguatge de la Universitat Pompeu Fabra. És membre del grup IULATERM des del 2003, i el seu àmbit de recerca és el processament del llenguatge natural i el desenvolupament de recursos, àrees en què ha liderat més de trenta projectes de recerca, la majoria amb finançament europeu, en àrees com la traducció automàtica, la recuperació d'informació, la classificació de documents i la producció automàtica de recursos lingüístics. Ha publicat més d'un centenar d'articles en revistes i congressos de prestigi internacional en l'àmbit de la lingüística computacional i la intel·ligència artificial. Darrerament s'ha centrat a avaluar el coneixement lingüístic que adquireixen els grans models del llenguatge i l'impacte de la quantitat i qualitat de les dades d'entrenament als resultats en tasques de classificació automàtica.

Resumen

Para describir el funcionamiento de los modelos de lenguaje se utilizan a menudo expresiones antropomorfistas que evitan los detalles técnicos, pero que provocan expectativas sobredimensionadas. El objetivo de este artículo es aportar una explicación realista de su funcionamiento y de 25 términos que se suelen utilizar en la descripción de esta tecnología.

PALABRAS CLAVE: *modelos grandes de lenguaje; inteligencia artificial generativa; terminología*

Resum

Un vocabulari bàsic dels grans models del llenguatge: una explicació i 25 termes

Per descriure el funcionament dels models de llenguatge sovint s'utilitzen expressions antropomorfistes que eviten els detalls tècnics, però que provoquen expectatives sobredimensionades. L'objectiu d'aquest article és aportar una explicació realista del seu funcionament i de 25 termes que se solen utilitzar en la descripció d'aquesta tecnologia.

PARAULES CLAU: *grans models de llenguatge; intel·ligència artificial generativa; terminologia*

Abstract

A basic vocabulary of large language models: An explanation and 25 terms

To describe the functioning of language models, anthropomorphic expressions are often used that avoid technical details, but lead to overstated expectations. The aim of this article is to provide a realistic explanation of how it works and of 25 terms that are often mentioned in description of this technology.

KEYWORDS: *large language models; generative artificial intelligence; terminology*

TERMINÀLIA 31 (2025): 27-39 · DOI: 10.2436/20.2503.01.223
Data de recepció: 4/03/2025. Data d'acceptació: 2/04/2025 amb modificacions
ISSN: 2013-6692 (impresa); 2013-6706 (electrònica) · *https://terminalia.iec.cat*

1 Introducción

La popularidad de los modelos de lenguaje, o, mejor dicho, de los sistemas que se hacen llamar **modelos grandes de lenguaje**[1] (LLM, del inglés *large language models*), es incuestionable, y se está difundiendo rápidamente la idea de que su uso pronto nos ayudará (o substituirá) en tareas como la creación de contenidos para *marketing*, en trabajos de educación y aprendizaje, o de asistentes para atención al cliente, y para asesoramiento, médico o legal. Si tan importante va a ser su impacto, es importante también que se entiendan las bases de su funcionamiento. Pero en la comunicación de sus sorprendentes capacidades se recurre a menudo a metáforas e imágenes antropomorfistas. Dicen que los LLM *interpretan*, *entienden* o *comprenden* las preguntas e instrucciones de los usuarios; que *razonan* o que *alucinan* cuando responden.[2] Con esta manera de hablar se evita describir su funcionamiento, que así pasa por ser inextricable, y se crean expectativas sobre la supuesta inteligencia de esta tecnología.

El objetivo de este artículo es explicar de forma intuitiva, con los mínimos detalles técnicos, el funcionamiento de los LLM y aportar un vocabulario básico, con explicaciones para no especialistas, y, sobre todo, veraz. No hemos pretendido definir cada término del vocabulario, pero sí ilustrar el contexto en el que se usa y describir a qué hace referencia. Esperamos que este vocabulario pueda ayudar a lingüistas, profesores, periodistas y traductores, entre otras personas interesadas en la tecnología, a expresarse con conocimiento y, sobre todo, sin crear expectativas ni sobredimensionar las capacidades de los LLM con referencias a comportamientos hasta hace poco solamente atribuibles a las personas y que no está demostrado que correspondan a lo que hacen estas máquinas.

En lo que sigue, se presenta una explicación del funcionamiento de los LLM basada en las publicaciones académicas disponibles en abierto. Algunos de los modelos de lenguaje más conocidos no han hecho públicos todos los detalles de sus sistemas (Metz, 2024), pero, aun así, se puede recurrir a los primeros trabajos que sus desarrolladores publicaron para cada innovación tecnológica y en los que sí se exponen los detalles técnicos. Esta decisión nos obliga a adoptar una perspectiva histórica y explicar cómo se van añadiendo innovaciones desde los primeros modelos generativos (GPT, del inglés *generative pre-trained transformer*) a los actualmente llamados LLM. Luego se presenta una lista de términos frecuentes en las descripciones y publicaciones de los LLM, normalmente en inglés. Hemos recogido o proponemos una versión en castellano para ellos y los señalamos en negrita en el texto. El criterio de selección de estos términos ha sido práctico. Son aquellos que solamente hemos mencionado, casi sin más explicaciones, en la descripción básica de cómo funciona la tecnología, que constituye la primera parte del artículo. El vocabulario ofrece entonces más información y la correspondencia con el término en inglés.

2 Del modelo de lenguaje a los modelos grandes de lenguaje

2.1 Un modelo de lenguaje

A grandes rasgos, un modelo de lenguaje es una máquina calculadora de probabilidades (Shannon, 1951). Un modelo de lenguaje calcula la probabilidad que tiene cada una de las palabras del vocabulario de ocupar una posición en una secuencia de otras palabras. Por ejemplo, de la secuencia i), el modelo calcularía que, de todas las palabras del vocabulario y después de esas 14 palabras (y de la coma), las más probables para la posición xxxx serían *doctor*, *médico*, *hospital*. Y serán muy poco probables palabras como *jardín* o *mar*.

> i) Esta noche pasada me he encontrado mal, así que al despertarme he llamado al xxxx
> ii) P(xxxx | Esta, noche, pasada, me, he, encontrado, mal...)

Para cada secuencia de palabras de entrada, el modelo de lenguaje calcula la probabilidad que tienen de aparecer todas y cada una de las palabras que tiene en el vocabulario. Se llama *entrenamiento* al cálculo de estas probabilidades, y se pueden calcular contando las veces que cada palabra efectivamente ha aparecido detrás de otra en grandes cantidades de textos. Los datos de entrenamiento son textos: secuencias de millones de palabras, espacios y signos de puntuación, y decimos que el LLM ha sido entrenado para la tarea de predecir la siguiente palabra. Así que, cuando se habla de que los LLM generan textos con errores factuales, o que **alucinan** para referirse a que el texto generado no está relacionado con lo que ha solicitado el usuario, en realidad, lo que sucede es que nada puede asegurar que una secuencia de palabras muy probables tenga que referirse a algo que sea verdadero. No obstante, cuantas más palabras haya en la secuencia que es el *input* para generar la siguiente más probable, más probabilidades hay de que se genere un texto correcto desde todos los puntos de vista: lingüístico y factual.

La tarea de predecir la siguiente palabra es simple, pero la tecnología desarrollada para que se puedan hacer los millones de cálculos que son necesarios para computar la probabilidad de las palabras a partir de millones de secuencias es un gran logro de la ingeniería del *software* (Vaswani *et al.*, 2017). Pero, la **generación automática de texto**, basada en calcular la palabra más probable, una detrás de otra, es la única tarea que puede resolver un modelo de lenguaje como tal (McCoy *et al.*, 2023).

2.2 LLM que resuelven una tarea

Para hacer que un LLM pueda resolver otras tareas es necesario disponer de unos textos específicos de entrenamiento, de datos que relacionen una secuencia de palabras y el resultado deseado según la tarea. Por ejemplo, una tarea muy popular es el análisis de opinión (también llamado *análisis de sentimientos*). Se trata de entrenar una máquina para que clasifique una secuencia de palabras según si expresa un sentimiento positivo o negativo (Pang *et al.*, 2002).

Por ejemplo, para las secuencias de palabras en iii)-v), el sistema calculará que algunas palabras de la secuencia «Me ha gustado mucho» y la palabra *positiva* están relacionadas, porque habrá muchos ejemplos que tengan ambas palabras, es decir que la palabra *positiva* es probable como continuación de esas anteriores. Es más, calculará que *me*, *ha* o *mucho* aparecen también con ejemplos de *negativa*, y que, en cambio, *gustado* aparece casi únicamente con secuencias en que está la palabra *positiva* y que, por tanto, para una secuencia nueva, la palabra *gustado* tendrá mayor peso, es decir, contribuirá más que las otras a que la palabra siguiente más probable sea *positiva*.

iii) Me ha gustado mucho la película es positiva.
iv) Me ha gustado mucho el jamón ibérico es positiva.
v) Odio el futbol es negativa.

Los datos específicos para esta tarea de análisis de opinión suelen venir de foros de discusión de medios sociales. También se pueden generar, por ejemplo, convirtiendo las estrellas de una página de recomendaciones en la secuencia de palabras «es positiva» o «es negativa». Con estos datos específicos, frases como las de los ejemplos iii)-v), se hace un entrenamiento adicional, un **reajuste** del modelo entrenado con textos genéricos, ahora llamado **modelo de lenguaje preentrenado** (Radford *et al.*, 2018) o **modelo fundacional** (Bommasani *et al.*, 2021), que es reajustado para convertirse en un **clasificador** que resuelve la tarea del análisis de opinión: para una nueva frase como vi) el sistema calcula ahora únicamente si *positiva* es más probable que *negativa*.

vi) Me ha gustado mucho el café.

Se puede calcular la clase más probable a la que pertenece una secuencia de palabras a partir de los datos usados para **reajustar**. Por ejemplo, hay conjuntos de datos para asignar la etiqueta *tóxico*, como el que se muestra en la figura 1.

Si queremos un LLM que resuelva una tarea como la de encontrar la respuesta a preguntas, habrá que reajustar un modelo preentrenado con textos generales con datos específicos de pregunta-respuesta como el ejemplo que se puede ver en la figura 2. Es un ejemplo de uno de los conjuntos de datos más usados: NaturalQuestions[4] de Google (Kwiatkowski *et al.*, 2019). Se trata de una compilación de preguntas que se le han hecho al buscador, y la respuesta que se podría encontrar en Wikipedia. NaturalQuestions contiene más de 325.000 preguntas y sus respuestas (en versión larga y corta).

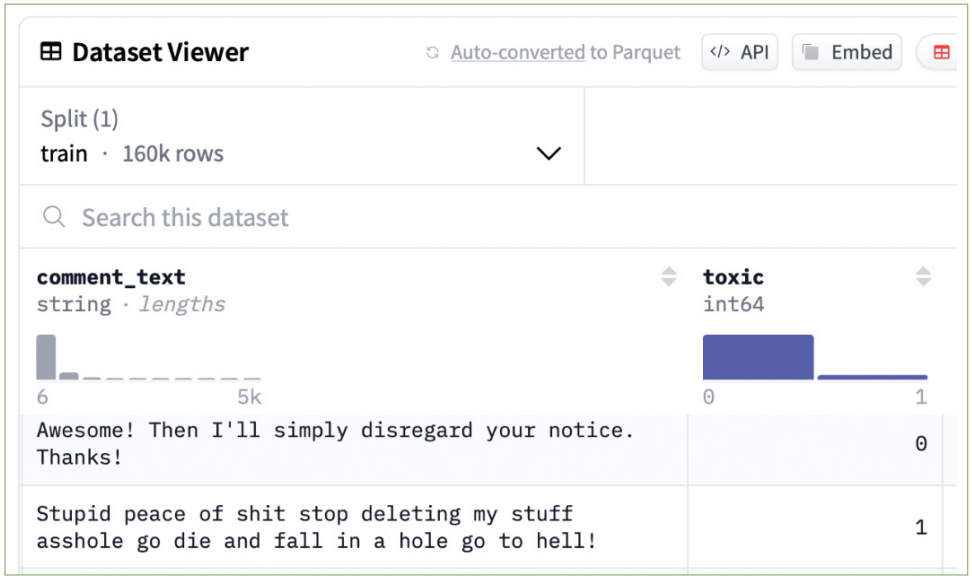

FIGURA 1. Captura de pantalla de los datos «jigsaw_toxicity» (Sileo, 2024) en el visor de *huggingface.com*.3 Contiene 160.000 comentarios de una red social con la anotación para reajustar un LLM como un clasificador de mensajes tóxicos (1) o no (0)

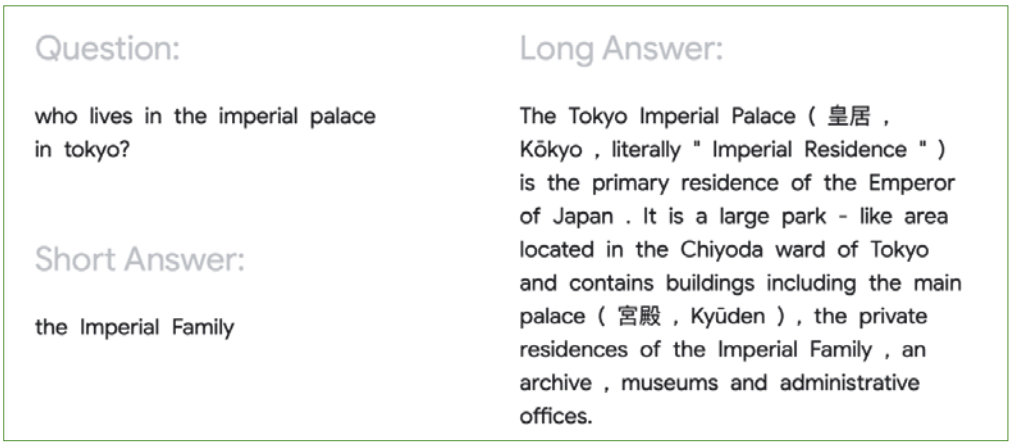

FIGURA 2. Ejemplo del conjunto de datos NaturalQuestions. Muestra la pregunta «Question», la respuesta larga asociada (*long answer*) y la respuesta corta (*short answer*). Fuente: *https://ai.google.com/research/NaturalQuestions*

2.3 LLM para más de una tarea

Reajustar un modelo preentrenado para cada tarea era efectivo, pero Radford *et al*. (2019) se plantearon por primera vez la posibilidad de calcular algo más que la probabilidad del *output* (ser una frase negativa o positiva) dado un *input*. Un LLM podía calcular la probabilidad del *output*, dado el *input* y también dada la tarea: P(output|input, tarea), porque en los textos también se suele encontrar esa información: la tarea, el *input* y el *output*, como una secuencia de palabras. Por ejemplo vii) o viii), pero también ix) o x), que parecen ser tareas que requieren la comprensión de la frase.

> vii) Di qué sentimiento expresa «me gusta el café», positivo.
> viii) Contesta la pregunta ¿cuál es la capital de Francia? París.
> ix) La traducción al inglés de «Me gusta el café» es «I like coffee».
> x) Calcula la suma de los siguientes números: 20, 5, 89 y 3? 117.

De esta forma, ya no sería necesario un entrenamiento adicional, el reajuste, con el que el sistema aprenda una tarea específica. Todas las tareas tendrían el mismo objetivo no específico: calcular la probabilidad de las secuencias de palabras. Pero, de nuevo, se necesitan datos de entrenamiento en los que tienen que estar explícitamente la tarea, el *input* y el *output*, para luego calcular la respuesta como la palabra más probable, una detrás de otra, a continuación de lo que el usuario describe como tarea en su **comando** (también llamado **instrucción** o *prompt*). Además, el modelo de lenguaje tenía que ser capaz de tener en cuenta muchas más palabras en el *input* como contexto condicionante para dar con la respuesta correcta. GPT2, el modelo de Radford *et al*. (2019), amplió este contexto de 512 a 1.024 palabras (o, mejor dicho, unidades o tokens: puntuación, símbolos, **subpalabras**, etc. identificadas en un proceso anterior de **tokenización**); en versiones posteriores se ha ampliado hasta 2.048. El LLM genera iterativamente palabras probables condicionadas por las palabras que hay en el comando, así que, cuantas más palabras, la respuesta estará más relacionada y será, probablemente, mejor. Los sistemas de diálogo, como ChatGPT, por defecto añaden los comandos anteriores y las respuestas dadas a cada nuevo comando del usuario para condicionar aún más la probabilidad de lo que tiene que generar y acertar con la respuesta (Wolfram, 2023).

2.4 El tamaño de los LLM

Para ver por qué se describen los diferentes modelos según su tamaño repasaremos de forma muy rápida qué es una red neuronal. Una red neuronal es un entramado de pequeñas unidades de computación, a las que llaman *neuronas*, por analogía con las células del cerebro que inspiraron esta tecnología de aprendizaje automático (Mitchell, 1997). Son unidades de computación porque cada una de ellas recibe un número de valores de *input*, y produce un único valor como output. En una red neuronal, la información que es el *output* de un nivel o capa de procesamiento es el *input* del nivel siguiente (figura 3), creándose así un entramado de relaciones de *input*/*output* entre las neuronas de la red que, por ejemplo, van procesando en paralelo las diferentes palabras de una frase y acaban diciendo si esta es positiva o negativa.

Estas conexiones entre neuronas se conocen como *parámetros* y son ponderables: puede haber y hay *inputs* que acaban teniendo más peso en el resultado que produce la neurona. Los modelos de lenguaje actuales, llamados **transformador** (*transformer*) y basados en el mecanismo de **atención** (Vaswani *et al*., 2017), se miden por el número de estos parámetros y de ahí el nombre: son modelos muy grandes.

Entrenar un LLM neuronal es comparar lo que produce con lo que tendría que haber producido según los

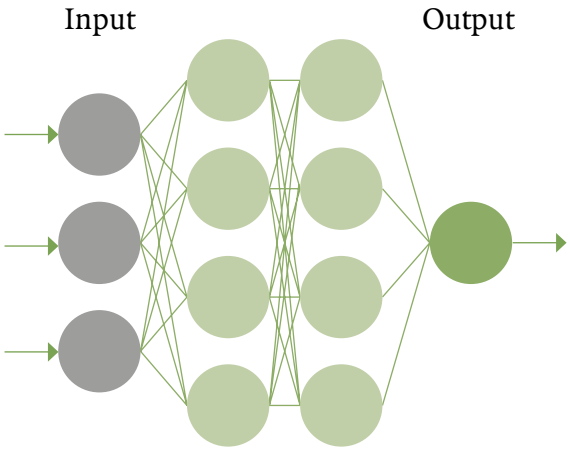

FIGURA 3. Representación esquemática de una red neuronal. A partir de un input de varios elementos (podrían ser palabras), se obtiene un resultado único (podría ser *positiva*). Fuente: Elaboración propia.

datos de entrenamiento. El LLM ha de calcular el peso de los parámetros para que acabe produciendo lo mismo que hay en los datos. Simplificando, y como hemos visto antes, cuando un modelo de lenguaje se entrena para predecir si la palabra *positiva* es la más probable, acabará calculando que *gustado* ha de tener más peso que otras, porque hay más veces que aparecen ambas en la misma secuencia.

El tamaño del modelo se convirtió en una característica importante cuando Brown *et al.* (2020) lanzaron GPT3 con 175.000 millones de parámetros y aumentaron el tamaño del contexto que condicionaba la probabilidad de la siguiente palabra. Para sus pruebas fueron 2.048 unidades, y en sus experimentos probaron que GPT3 podía realizar más tareas y con mejores resultados solamente usando el *input* del usuario, sin ningún aprendizaje o reajuste específico. Estaban mostrando el inicio a un LLM de propósito general, sin necesidad de especialización ni de datos especializados.

Brown *et al.* (2020) y su GPT3 fueron un hito importante en el desarrollo de los LLM por aumentar su tamaño, de 1.500 millones de GPT2 a 175.000 millones de parámetros, y también por aumentar el tamaño de los datos de entrenamiento, ya que estudios anteriores (Kaplan *et al.*, 2020) habían demostrado que el número de parámetros no era la única condición para obtener buenos resultados. El aumento del tamaño del modelo tenía que ir acompañado de más datos y entrenaron GPT3 con 300.000 millones de palabras, provenientes de textos de Internet, colecciones de libros y páginas de Wikipedia.[5] GPT2 había sido entrenado con 40.000 millones.

El cambio de escala en tamaño y datos se presentaba como la justificación de los resultados obtenidos para las diferentes tareas que GPT3 respondía relativamente bien, teniendo en cuenta que no había sido reajustado con datos específicos para cada tarea, por lo menos que ellos supieran. Los mismos autores se avanzaban a posibles sospechas de contaminación y memorización. Por *contaminación*, se referían a que, con tantos datos de entrenamiento, era posible que el LLM hubiera sido entrenado con textos que contenían las soluciones de los conjuntos de prueba que estaban usando para evaluarlo. Por *memorización*, se referían a que era posible que el modelo registrara secuencias enteras de palabras. Por ejemplo, el LLM podía responder correctamente la pregunta «¿cuánto son 2 + 2?», porque es una secuencia frecuente que suele aparecer con el resultado, pero no «¿cuánto son 2 + 2,685?», una secuencia más rara. Aunque aportaron datos que minimizaban el posible alcance de la memorización, parte de la comunidad científica siguió dudando de las capacidades de aprendizaje de GPT3; por ejemplo, el artículo de Bender *et al.* (2021), en el que se comparaba GPT3 con un loro que solamente repetía el texto con el que había sido entrenado, se hizo muy popular.

Por último, pero muy importante, Brown *et al.* (2020) también advertían que, al generar texto seleccionando la palabra más frecuente dada una secuencia de otras palabras, GPT3 reproducía también secuencias de palabras con **sesgos** de género, contenidos racistas y lenguaje ofensivo.

3 Datos para seguir las instrucciones del usuario

En la búsqueda de un LLM de propósito general y que evitara los sesgos y otros defectos en la calidad y los contenidos de los resultados, Ouyang *et al.* (2022) proponen crear un **modelo instruido**: reajustar un gran, gran LLM como GPT3 con textos que contuvieran explícitamente cómo responder a una tarea y su solución. Así, el objetivo del sistema dejaba de ser predecir la siguiente palabra para ser, en sus propias palabras, «follow the user's instructions helpfully and safely» (seguir las instrucciones de los usuarios de forma útil

y segura). Lo presentaron como **alinear** los LLM con la intención de los usuarios.

Encomendaron el desarrollo de los nuevos datos a un grupo de cuarenta personas a las que encargaron dos trabajos. En primer lugar, les pidieron que escribieran el texto que describía la tarea que querían que el sistema llevara a cabo, y también los textos que eran el resultado deseado. Por ejemplo, «Dime el sentimiento de este texto, bla, bla, bla» y el resultado «negativo». Con estos datos, se reajustó GPT3, y se volvió a utilizar reajustado para generar automáticamente respuestas a los comandos que habían escrito los humanos.

En segundo lugar, pidieron al equipo de humanos que puntuaran esas respuestas generadas automáticamente de mejor a peor. Los humanos tenían unas instrucciones que describían lo que se tenía que considerar un buen resumen, o una buena traducción, por ejemplo, y también si era texto ofensivo o si contenía sesgos. Finalmente, con estos nuevos datos que relacionaban textos y preferencias humanas, entrenaron otro componente de aprendizaje automático llamado **aprendizaje por refuerzo**, que reajustaba el GPT3 anterior ahora solamente para aprender a seleccionar, de entre los diferentes resultados que se generaban, los que habrían obtenido mayor puntuación humana. Este sistema era InstructGPT (Ouyang *et al.*, 2022), el prototipo de ChatGPT. La evaluación de los resultados de este modelo, ahora reajustado con información de las preferencias humanas y comparándolo con los resultados de GPT3 sin instruir, confirmó que InstructGPT generaba textos más seguros y que cubría más tareas. También se habló de **propiedades emergentes** (Wei *et al.*, 2022) para nombrar la capacidad de realizar tareas para las que no había sido reajustado específicamente: sumas y restas, traducción, respuesta a preguntas, inferencias o corrección gramatical.

Pero InstructGPT también cometía errores que describían los mismos autores. Cuando el sistema recibía un comando con una premisa falsa, el LLM asumía que esa premisa era cierta. Podía producir más de una opción, aunque por el contexto dado solamente hubiera una. La respuesta era peor cuantas más restricciones se ponían a la respuesta, como por ejemplo hacer un resumen de un número determinado de líneas. Y, como contenía muy pocos datos en otras lenguas además del inglés, el sistema contestaba en inglés, aunque el *input* estuviera en otra lengua. Estos son errores en los que todavía incurren la mayoría de los LLM disponibles, aunque con menos frecuencia para aquellos que más han aumentado la cantidad de datos con los que han sido instruidos. Además, algunos LLM ofrecen un parámetro llamado **temperatura**, que puede ajustar el usuario, para que se generen textos más o menos creativos, es decir, secuencias de palabras más probables y, por tanto, más parecidas a los datos de entrenamiento.

Con el objetivo de aumentar las capacidades de los diferentes modelos en cuanto a tareas y lenguas en las que se pueden expresar, en los últimos años se han ido construyendo nuevos conjuntos de datos de instrucciones con los que reajustar los modelos. Entre las tareas para las que se están generando rápidamente datos textuales con los que reajustar los modelos están datos de operaciones aritméticas[6] y de problemas matemáticos que se han copiado de libros y manuales de todos los cursos[7] (figura 4). También hay datos de código escrito en diferentes lenguajes de programación[8] con los

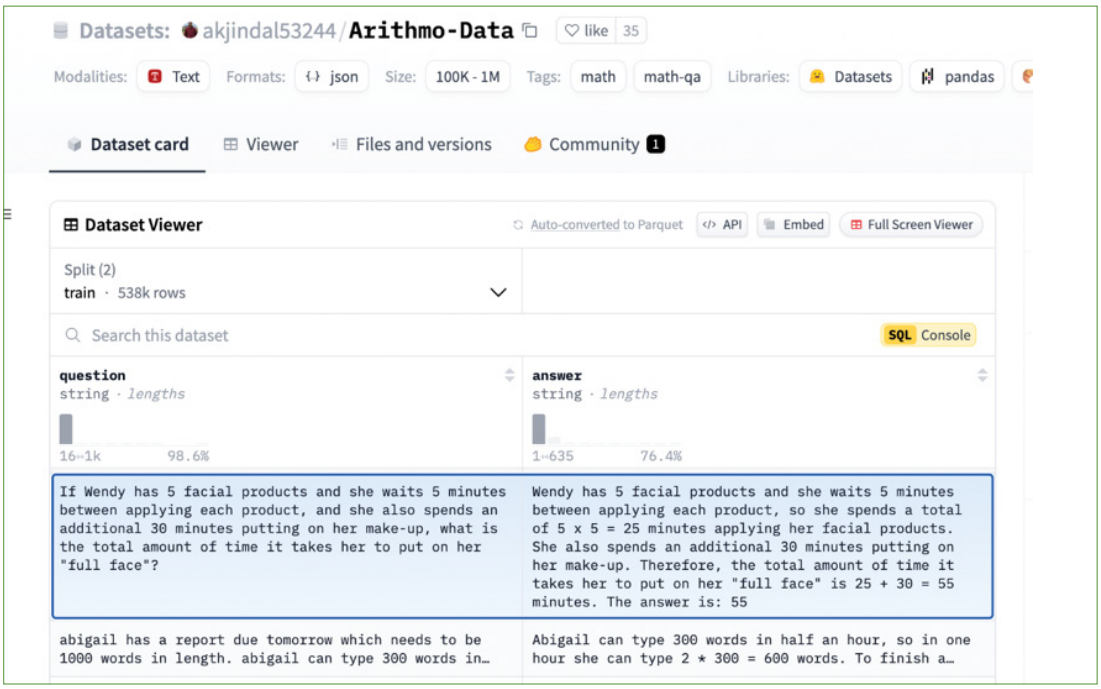

FIGURA 4. Ejemplo de los datos MAmmoTH (Yue et al. 2023), con problemas de matemáticas y sus respuestas. Captura de pantalla del visor de *huggingface.co*.

que convertir un LLM en un asistente de programador. Otros ejemplos que muestran la variedad y el tamaño de estos conjuntos de datos son los siguientes: Flan 2022 (Longpre *et al.*, 2023) son 15 millones de ejemplos de instrucciones para 1.836 tareas diferentes; Aya (Singh *et al.*, 2024) son 503 millones de instrucciones, en 114 lenguas, para 12 tareas, entre las que se incluyen respuesta a preguntas, resumen automático, traducción, paráfrasis, análisis de opinión, inferencia en lenguaje natural; Zhang *et al.* (2024) ofrece una compilación actualizada de LLM junto con los datos de instrucción que han usado, aunque solamente los que se han hecho públicos.

Si los buenos resultados de la tecnología actual de los LLM dependen críticamente de las enormes cantidades de datos para entrenar y reajustar que se han usado, se plantea la pregunta de si pueden resolver nuevas tareas propuestas por usuarios o si sus resultados están condicionados a tareas para las que han sido entrenados. Se habla de **aprendizaje en contexto** o **aprendizaje por transferencia** para referirse a la supuesta capacidad de los LLM de utilizar el preentrenamiento y las instrucciones recibidas en el reajuste para dar respuesta a una nueva tarea propuesta por el usuario si en el comando (o *prompt*) se describe adecuadamente la tarea y se aportan unos ejemplos para demostrar cómo ha de ser el resultado (Brown *et al.*, 2020). Usar un comando con una descripción adecuada de la tarea y algunos ejemplos que contengan la solución mejora las respuestas del LLM, aunque los ejemplos no afecten ya las probabilidades que el LLM haya calculado en el entrenamiento o reajuste del modelo. Se habla de **ingeniería de los comandos** (o *prompt engineering*) para describir la heurística que conlleva encontrar instrucciones óptimas para resolver las tareas, y de la forma propuesta por el usuario como **comandar con cero, uno o varios ejemplos**. Min *et al.* (2022) sugieren que las palabras que se introducen en el comando actuarán como condicionantes añadidos que restringirán más las palabras probables, facilitando así que el LLM dé con la respuesta correcta.

Por último, para hacer que los LLM fueran un producto usable por el gran público, sus desarrolladores incorporaron componentes que limitaran la posibilidad de que produjeran información falsa, contenido indeseable, texto tóxico o que contenga sesgos o simplemente que la respuesta no corresponda en ningún sentido a lo que se ha pedido. Son los llamados **guardarraíles**, que monitorizan los textos que el sistema recibe como *input* o los que los LLM producen. Pueden ser programados para aceptar el comando como es, o modificarlo troceándolo para que se procesen tareas complejas en pasos lógicos parciales (como una **cadena de acciones**) o incluso rechazar tanto el comando como el texto producido por el LLM. Ya hay empresas especializadas en desarrollar estos componentes adicionales, que, en último término, garantizan la usabilidad de los LLM en aplicaciones profesionales. Muy recientemente, y para garantizar que el texto generado sea correcto factualmente, también se han incorporado como componentes adicionales bases de datos de información factual que se compara con las secuencias producidas por el LLM en una técnica llamada **generación aumentada por recuperación de información** (RAG, del inglés *retrieval-augmented generation*), o se combina el resultado de la generación con los datos encontrados en la web con un buscador normal.

4. Vocabulario: 25 términos útiles

Los términos seleccionados han sido mencionados en el texto de las secciones anteriores, y se han señalado en negrita.

4.1 Alucinar, alucinación (*hallucination*)

Se habla de *alucinación* cuando un LLM genera una secuencia de palabras correcta lingüísticamente, pero que es incorrecta factualmente (lo que describe no es cierto) o bien no tiene ninguna relación con el *input*.

4.2 Alinear, alineación (*alignment*)

Reajustar un modelo para que produzca resultados que estén en consonancia con las preferencias humanas, a menudo mediante el método de aprendizaje por refuerzo. Los datos con los que se hace la alineación se consiguen con informantes humanos a los que, entre otras tareas, les han pedido que ordenen diferentes textos generados por un LLM según sus preferencias con respecto a diferentes parámetros: contenido ofensivo, calidad del resumen del texto, etc.

4.3 Aprendizaje en contexto (*in-context learning*, ICL)

Se trata de la técnica de incluir en el comando la descripción de la tarea y un número de ejemplos con la solución para que el LLM infiera cómo realizar esa nueva tarea. También llamado *aprendizaje por transferencia* o *aprendizaje por instrucciones* (Brown *et al.*, 2020).

4.4 Aprendizaje por refuerzo (*reinforcement learning*)

Es un método de aprendizaje automático en el cual un programa evalúa la eficacia de otro programa con respecto a un objetivo previamente definido y calcula las acciones que este otro programa ha de realizar para conseguir el objetivo óptimamente. El aprendizaje por refuerzo es el método que se ha utilizado en ChatGPT para alinear los resultados generados por un LLM con las preferencias humanas, específicamente, con respecto a no producir resultados cuyo contenido pueda ser nocivo: sexual, de inducción al odio, al racismo, etc.

4.5 Aprendizaje por transferencia (*transfer learning*)

Es un método de aprendizaje automático con LLM con el cual se entrena con un conjunto de datos específicos de una tarea de gran tamaño, para posteriormente hacer un reajuste con una cantidad pequeña de otro conjunto de datos para una tarea parecida, y se espera que los primeros datos ayuden a resolver la tarea con los nuevos datos. Uno de los ejemplos más típicos es el de resolver tareas multilingües: se entrena con muchos datos en inglés y se reajusta con unos pocos datos en otra lengua, por ejemplo, español, para que el LLM haga la tarea también con *input* en esta otra lengua.

4.6 Atención (*attention*)

Es el mecanismo básico del transformador y es el encargado de combinar los vectores que representan las diferentes unidades (palabras, subpalabras, símbolos, etc.) de una misma secuencia para construir una representación también vectorial de cada una de ellas que contenga información sobre las otras. Por este motivo se habla de vectores contextuales que representarían diferente la palabra *banco* si aparece en una secuencia con la palabra *dinero* de si aparece con la palabra *peces* (Vaswani *et al.*, 2017).

4.7 Cadena de acciones (*chain of thought prompting*)

Un comando con una cadena de acciones es la que incluye una serie de pasos intermedios razonados que llevan a un resultado final (Wei *et al.*, 2022).

4.8 Clasificador (*classifier*)

Sistema automático que, dadas unas clases predefinidas, es entrenado para asignar a un *input* nuevo una de esas clases.

4.9 Comando, instrucción (*prompt*)

Mensaje que introduce el usuario como *input* en una interacción con un sistema LLM. Puede contener texto que describe la tarea, y también ejemplos como demostraciones de cómo es la respuesta deseada. En el texto se ha traducido el término original inglés por *comando* como instrucción que se da a una máquina. Hernández Fernández y Ferrer i Cancho (2023) aportan una interesante discusión sobre la terminología de la inteligencia artificial, y en particular sobre los orígenes del término *prompt*.

4.10 Comandar con cero, uno o varios ejemplos (*zero, one o few-shot prompting*)

En la interacción con el sistema, el usuario incluye en el comando la descripción de la tarea y una o algunas demostraciones de la tarea resuelta. Se dice entonces que la información sobre la solución de tareas adquirida en el preentrenamiento o en los reajustes se transfiere a la nueva tarea: por ejemplo, clasificar frases en nuevas clases especificadas por el usuario aprovecharía el aprendizaje del análisis de opinión. Los datos provistos por el usuario, que pueden incluir de cero a varios ejemplos, pueden ser de hasta 2.048 unidades (palabras, signos, etc.); no causarán ningún cambio en los pesos de los parámetros calculados en el entrenamiento o en posibles reajustes, pero limitarán aún más la probabilidad de las siguientes palabras posibles. También se ha llamado *in context learning* (ICL).

4.11 Generación automática de texto (*text generation*)

Tarea en la que, a partir de un *input* que pueden ser datos o comandos, un programa genera un texto que contiene los datos o que cumple con las instrucciones dadas.

4.12 Generación aumentada por recuperación de información (*retrieval-augmented generation*, RAG)

Para mejorar el contenido factual de las respuestas de los LLM, esto es que se produzcan respuestas correctas en lo que se refiere a hechos, en estos sistemas RAG se utiliza una base de datos de conocimiento indexada, y el contenido se representa como vectores (*embeddings*). Esto permite llevar a cabo una búsqueda «semántica»: encontrar los contenidos cuyo vector sea el más similar al de la pregunta que se ha hecho al sistema. Una vez identificado, el contenido se pasa al LLM como input para que genere el texto final de la respuesta.

4.13 Guardarraíles (*guardrails*, *safeguard models*)

Se trata de clasificadores entrenados específicamente para identificar textos que contengan referencias sexuales, de odio o violencia, acoso, autoagresiones, etc. Estos clasificadores actúan para detectar tanto comandos incorrectos de usuarios, como resultados del LLM que contengan sesgos o lenguaje ofensivo.

4.14 Ingeniería de los comandos (*prompt engineering*)

Es el proceso de buscar heurísticamente el comando que parece provocar una mejor respuesta a la tarea que el usuario pretende que el LLM lleve a cabo.

4.15 Modelo de lenguaje (*language model*)

Estimación de la probabilidad de que aparezca una palabra (o frase, o símbolo, o carácter) en un texto, dadas las palabras anteriores. Las probabilidades pueden utilizarse para estimar la corrección o carácter de normalidad de una secuencia de palabras en una lengua. Los **modelos grandes de lenguaje** (*gran modelo de lenguaje* o *modelo de lenguaje extenso*) estiman la probabilidad de una palabra (o subpalabra), distribuyéndola entre todas las de un vocabulario de palabras limitado y proponiendo como palabra siguiente a generar la que tiene más probabilidad.

4.16 Modelo de lenguaje preentrenado (*pre-trained language model*)

Se habla de *modelo preentrenado* cuando el modelo se ha desarrollado desde cero: inicialización de los parámetros, construcción del modelo y entrenamiento con una gran cantidad de texto. Una vez disponible, es la base (por eso también se ha llamado *modelo base*) para ajustar modelos especializados en una o diferentes tareas, o para, mediante un reajuste con instrucciones, convertirlo en un modelo instruido.

4.17 Modelo fundacional (*foundational model*)

Es otra forma de referirse a un modelo de lenguaje preentrenado. Con este nombre, Bommasani *et al.* (2021) se referían a un modelo que es el fundamento para construir modelos especializados, un elemento necesario, pero no suficiente, para desarrollar aplicaciones.

4.18 Modelo instruido (*instruction-tuned model*)

Se llama así a un modelo de lenguaje preentrenado, base o fundacional, que ha sido reajustado con datos que contienen explícitamente las instrucciones de una tarea y su solución.

4.19 Propiedades emergentes (*emergent capacities*, *properties*)

Los primeros en usar este término en el ámbito de los LLM fueron Wei *et al.* (2022) para referirse al hecho de que los modelos grandes podían resolver tareas que modelos más pequeños eran incapaces de resolver. Luego, se ha usado también para hablar de tareas que los LLM parecen poder hacer sin haber sido entrenados específicamente para aprenderlas. Las tareas que pusieron como ejemplos fueron el cálculo aritmético o la respuesta a preguntas. Hasta la fecha no hay consenso sobre el hecho de que estas capacidades de los modelos grandes sean efectivamente emergentes (Schaeffer *et al.*, 2023).

4.20 Reajuste, reajustar (*fine-tuning*)

Proceso de añadir una nueva capa de neuronas a un modelo preentrenado o fundacional y llevar a cabo un entrenamiento adicional para que el LLM calcule probabilidades añadidas a partir de nuevos datos que pueden ser de textos de un ámbito diferente al que se ha usado para el entrenamiento o de una o más tareas específicas.

4.21 Sesgo (*bias*)

En los ensayos experimentales se habla de *sesgo* para referirse a una recopilación de datos en la que se da una sobrerrepresentación de unos ciertos eventos o valores de variables. En el contexto del aprendizaje automático, se habla de *sesgos* cuando, dado que no hay selección de textos, los textos recopilados contienen elementos sobrerrepresentados que convierten el texto generado por el sistema en políticamente incorrecto, ofensivo o tóxico.

4.22 Subpalabra (*subword*)

Los algoritmos de tokenización empleados actualmente segmentan las palabras en secuencias de caracteres que aparecen juntos con frecuencia; así las unidades que procesa el LLM pueden ser trozos de palabras que se llaman *subpalabras*. El objetivo de esta tokenización está relacionado con la capacidad limitada de los vocabularios de los LLM. Por cuestiones de complejidad de cálculo, los primeros LLM tenían un vocabulario de 32.000 unidades, con lo que era imposible recoger todas las palabras de una lengua. Las palabras se partían en diferentes tokens para cubrir más palabras de la lengua.

4.23 Temperatura (*temperature*)

Es un parámetro ajustable por el usuario que permite graduar la predicción de la siguiente palabra para que se prioricen las más probables (*temperatura baja*) o algunas menos probables (*temperatura alta*). Las palabras menos probables hacen que la frase sea menos común, así que subir la temperatura se entiende como una forma de aumentar la creatividad del LLM.

4.24 Tokenización (*tokenization*)

Dividir un texto en elementos de análisis, básicamente palabras, signos de puntuación, etc. Este paso se realiza antes del procesamiento de la red neuronal, es decir, de los cálculos de probabilidades de secuencias.

4.25 Transformador (*transformer*)

Vaswani *et al.* (2017) llamaron así a la arquitectura de red neuronal combinada con un mecanismo específico llamado *atención* que se ha convertido en la base de los modelos grandes de lenguaje. Los LLM actuales son

concretamente transformadores generativos preentrenados (*generative pre-trained transformers*), GPT (Radford *et al.* 2018), especializados en la generación de la palabra siguiente a partir de las anteriores, como hemos visto. En su planteamiento inicial querían reducir la cantidad de datos necesarios que otros métodos de aprendizaje automático requerían.

5 Conclusión

La tarea básica de los LLM es predecir la palabra más probable dada una secuencia de palabras. La forma en que lo consiguen es un alarde tecnológico de mucha complejidad que se ha ido presentando como el logro del largamente perseguido objetivo de crear máquinas que puedan entender el lenguaje natural. Entender a los usuarios sin necesidad de aprender un lenguaje de programación había sido el objetivo desde mediados del siglo XX, con diferentes propuestas que proponían que un programa de *software* podía responder al usuario como si entendiera o interpretara lo que este le pedía hablando. La clave estaba en el «como si», porque se pretendía simular una respuesta humana. El objetivo de hacer como si la máquina entendiera ha promovido el uso de metáforas y referencias antropomorfistas en el ámbito de la inteligencia artificial, que se replican y aumentan en los textos divulgativos.

Con la aparición de los LLM, con sus sorprendentes resultados, y a falta de que hicieran públicos detalles críticos de su diseño y entrenamiento, el uso de estas metáforas ha ido creando una cierta obscuridad sobre su funcionamiento. En este artículo nos proponíamos explicar el funcionamiento de los LLM de forma intuitiva, pero realista y veraz, y aportar un vocabulario básico que complemente y amplíe las explicaciones. Esperamos contribuir a una mejor comprensión sobre las capacidades de estos sistemas, de los que las predicciones dicen que cambiarán profundamente la forma de trabajar de los humanos y que nos complementarán en diferentes tareas, algunas críticas. También esperamos que, al conocer mejor su funcionamiento, las personas interesadas en su divulgación lo expresen de forma más clara para que se vaya evaluando el coste que estos desarrollos están teniendo: el tratamiento desigual para lenguas que no disponen de recursos textuales suficientes, una información insuficiente sobre los sesgos que se manifiestan en los textos producidos por los LLM, y también la existencia de componentes que pueden introducir o promocionar otros sesgos o mensajes no siempre de forma manifiesta.

6 Bibliografía

Bender, Emily M.; Gebru, Timnit; McMillan-Major, Angelina; Shmitchell, Shmargaret (2021). «On the dangers of stochastic parrots: Can language models be too big?». En: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FaccT'21)* [en línea]. Nueva York: Association for Computing Machinery, pp. 610-623. <https://doi.org/10.1145/3442188.3445922>.

Bommasani, Rishi; Hudson, Drew A.; Ehsan, Adeli; Altman, Russ; Arora, Simran; Arx, Sydney von; Bernstein, Michael S.; Bohg, Jeannette; Bosselut, Antoine; Brunskill, Emma; Brynjolfsson, Erik; Buch, Shyamal; Card, Dallas; Castellon, Rodrigo; Chatterji, Niladri S.; Chen, Annie S.; Creel, Kathleen A.; Davis, Jared; Demszky, Dora; Donahue, Chris; Moussa Koulako Bala Doumbouya; Durmus, Esin; Ermon, Stefano; Etchemendy, John; Ethayarajh, Kawin; Fei-Fei, Li; Finn, Chelsea; Gale, Trevor; Gillespie, Lauren; Goel, Karan; Goodman, Noah D.; Grossman, Shelby; Guha, Neel; Hashimoto, Tatsunori; Henderson, Peter; Hewitt, John; Ho, Daniel E.; Hong, Jenny; Hsu, Kyle; Huang, Jing; Icard, Thomas F.; Jain, Saahil; Jurafsky, Dan; Kalluri, Pratyusha; Karamcheti, Siddharth; Keeling, Geoff; Khani, Fereshte; Khattab, Omar; Wei Koh, Pang; Krass, Mark S.; Krishna, Ranjay; Kuditipudi, Rohith; Kumar, Ananya; Ladhak, Faisal; Lee, Mina; Lee, Tony; Leskovec, Jure; Levent, Isabelle; Lisa Li, Xiang; Li, Xuechen; Ma, Tengyu; Malik, Ali; Manning, Christopher D.; Mirchandani, Suvir; Mitchell, Eric; Munyikwa, Zanele; Nair, Suraj; Narayan, Avanika; Narayanan, Deepak; Newman, Benjamin; Nie, Allen; Niebles, Juan Carlos; Nilforoshan, Hamed; Nyarko, Julian F.; Ogut, Giray; Orr, Laurel J.; Papadimitriou, Isabel; Park, Joon Sung; Piech, Chris; Portelance, Eva; Potts, Christopher; Raghunathan, Aditi; Reich, Robert; Ren, Hongyu; Rong, Frieda; Roohani, Yusuf; Ruiz, Camilo; Ryan, Jack; R'e, Christopher; Sadigh, Dorsa; Sagawa, Shiori; Santhanam, Keshav; Shih, Andy; Parasuram Srinivasan, Krishna; Tamkin, Alex; Taori, Rohan; Thomas, Armin W.; Tramèr, Florian; Wang, Rose E.; Wang, William; Wu, Bohan; Wu, Jiajun; Wu, Yuhuai; Xie Sang, Michael; Yasunaga, Michihiro; You, Jiaxuan; Zaharia, Matei A.; Zhang, Michael; Zhang, Tianyi; Zhang, Xikun; Zhang, Yuhui; Zheng, Lucia (2021). «On the opportunities and risks of foundation models». *arXiv:2108.07258* [en línea]. <https://doi.org/10.48550/arXiv.2108.07258>. [Consulta: 04/03/2025].

BROWN, Tom B.; MANN, Benjamin; RYDER, Nick; SUBBIAH, Melanie; KAPLAN, Jared; DHARIWAL, Prafulla; NEELAKANTAN, Arvind; SHYAM, Pranav; SASTRY, Girish; ASKELL, Amanda; AGARWAL, Sandhini; HERBERT-VOSS, Ariel; KRUEGER, Gretchen; HENIGHAN, Tom; CHILD, Rewon; RAMESH, Aditya; ZIEGLER, Daniel M.; WU, Jeffrey; WINTER, Clemens; HESSE, Christopher; CHEN, Mark; SIGLER, Eric; LITWIN, Mateusz; GRAY, Scott; CHESS, Benjamin; CLARK, Jack; BERNER, Christopher; MCCANDLISH, Sam; RADFORD, Alec; SUTSKEVER, Ilya; AMODEI, Dario (2020). «Language models are few-shot learners». En: *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20)* [en línea]. Red Hook, NY: Curran Associates Inc., pp. 1877-1901. <https://doi.org/10.48550/arXiv.2005.14165>. [Consulta: 04/03/2025].

HERNÁNDEZ FERNÁNDEZ, Antoni; FERRER I CANCHO, Ramon (2023). «Lingüística quantitativa i lleis lingüístiques: de la lingüística a la intel·ligència artificial i la tecnoètica». *Terminàlia* [en línea], 27, pp. 72-79. <https://revistes.iec.cat/index.php/Terminalia/article/view/150577 >. [Consulta: 04/03/2025].

KAPLAN, Jared; MCCANDLISH, Sam; HENIGHAN, Tom; BROWN, Tom B.; CHESS, Benjamin; CHILD, Rewon; GRAY, Scott; RADFORD, Alec; WU, Jeff; AMODEI, Dario (2020). «Scaling laws for neural language models». *arXiv:2001.08361* [en línea]. <https://doi.org/10.48550/arXiv.2001.08361>. [Consulta: 04/03/2025].

KWIATKOWSKI, Tom; PALOMAKI, Jennimaria; REDFIELD, Olivia; COLLINS, Michael; PARIKH, Ankur; ALBERTI, Chris; EPSTEIN, Danielle; POLOSUKHIN, Illia; DEVLIN, Jacob; LEE, Kenton; TOUTANOVA, Kristina; JONES, Llion; KELCEY, Matthew; CHANG, Ming-Wei; DAI, Andrew M.; USZKOREIT, Jakob; LE, Quoc; PETROV, Slav (2019). «Natural questions: A benchmark for question answering research». *Transactions of the Association for Computational Linguistics* [en línea], 7, pp. 452-466. <https://doi.org/10.1162/tacl_a_00276>. [Consulta: 04/03/2025].

LONGPRE, Shayne; HOU, Le; VU, Tu; WEBSON, Albert; CHUNG, Hyung Won; TAY, Yi; ZHOU, Denny; LE, Quoc V.; ZOPH, Barret; WEI, Jason; ROBERTS, Adam (2023). «The flan collection: Designing data and methods for effective instruction tuning». En: *Proceedings of the 40th International Conference on Machine Learning (ICML'23)* [en línea], artículo 941, pp. 22631-22648. <https//dl.acm.org/doi/10.5555/3618408.3619349>. [Consulta: 04/03/2025].

MCCOY, R. Thomas; YAO, Shunyu; FRIEDMAN, Dan; HARDY, Matthew; GRIFFITHS, Thomas L. (2023). «Embers of autoregression: Understanding large language models through the problem they are trained to solve». *arXiv:2309.13638* [en línea]. <https://doi.org/10.48550/arXiv.2309.13638>. [Consulta: 04/03/2025].

METZ, Rachel (2024). «OpenAI and the fierce AI industry debate over open source». *Bloomberg. Newsletter Tech Daily* [en línea]. <https://www.bloomberg.com/news/newsletters/2024-03-15/openai-tumult-raises-question-of-how-open-an-ai-company-should-be>. [Consulta: 04/03/2025].

MIN, Sewon; LYU, Xinxi; HOLTZMAN, Ari; ARTETXE, Mikel; LEWIS, Mike; HAJISHIRZI, Hannaneh; ZETTLEMOYER, Luke (2022). «Rethinking the role of demonstrations: What makes in-context learning work?». En: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* [en línea]. Abu Dhabi: Association for Computational Linguistics. <https://doi.org/10.18653/v1/2022.emnlp-main.759>. [Consulta: 04/03/2025].

MITCHELL, Tom (1997). *Machine learning* [en línea]. McGraw-Hill Science/Engineering/Math. <https://www.cs.cmu.edu/ffitom/files/MachineLearningTomMitchell.pdf>. [Consulta: 04/03/2025].

OUYANG, Long; WU, Jeff; JIANG, Xu; ALMEIDA, Diogo; WAINWRIGHT, Carroll L.; MISHKIN, Pamela; ZHANG, Chong; AGARWAL, Sandhini; SLAMA, Katarina; RAY, Alex; SCHULMAN, John; HILTON, Jacob; KELTON, Fraser; MILLER, Luke; SIMENS, Maddie; ASKELL, Amanda; WELINDER, Peter; CHRISTIANO, Paul; LEIKE, Jan; LOWE, Ryan (2022). «Training language models to follow instructions with human feedback». En: *Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS'22)* [en línea]. Red Hook, NY: Curran Associates Inc., pp. 27730-27744. <https://doi.org/10.48550/arXiv.2203.02155>. [Consulta: 04/03/2025].

PANG, Bo; LEE, Lillian; VAITHYANATHAN, Shivakumar (2002). «Thumbs up? Sentiment classification using machine learning techniques». En: *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)* [en línea]. S. l.: Association for Computational Linguistics, pp. 79-86. <https://doi.org/10.3115/1118693.1118704>. [Consulta: 04/03/2025].

RADFORD, Alec; NARASIMHAN, Karthik; SALIMANS, Tim; SUTSKEVER, Ilya (2018). «Improving language understanding by generative pre-training». *OpenAI* [en línea], pp. 1-12. <https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf>. [Consulta: 04/03/2025].

RADFORD, Alec; WU, Jeff; CHILD, Rewon; LUAN, Davi; AMODEI, Dario; SUTSKEVER, Ilya (2019). «Language models are unsupervised multitask learners». *OpenAI Blog* [en línea]. <https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf>. [Consulta: 04/03/2025].

SILEO, Damien (2024). «Tasksource: A large collection of NLP tasks with a structured dataset preprocessing framework». En: *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)* [en línea]. Turín: ELRA and ICCL, pp. 15655-15684. <https://aclanthology.org/2024.lrec-main.1361/>. [Consulta: 04/03/2025].

SCHAEFFER, Rylan; MIRANDA, Brando; KOYEJO, Sanmi (2023). «Are emergent abilities of large language models a mirage?». *Advances in Neural Information Processing Systems* [en línea], 36 pp. 55565-55581. <https://proceedings.neurips.cc/paper_files/paper/2023/file/adc98a266f45005c403b8311ca7e8bd7-Paper-Conference.pdf>. [Consulta: 04/03/2025].

SHANNON, Claude E. (1951). «Prediction and entropy of printed English». *Bell System Technical Journal* [en línea], vol. 30, núm. 1, pp. 50-64. <https://doi:10.1002/j.1538-7305.1951.tb01366.x>. [Consulta: 04/03/2025].

SINGH, Shivalika; VARGUS, Freddie; D'SOUZA, Daniel; KARLSSON, Börje; MAHENDIRAN, Abinaya; KO, Wei-Yin; SHANDILYA, Herumb; PATEL, Jay; MATACIUNAS, Deividas; O'MAHONY, Laura; ZHANG, Mike; HETTIARACHCHI, Ramith; WILSON, Joseph; MACHADO, Marina; MOURA, Luisa; KRZEMIŃSKI, Dominik; FADAEI, Hakimeh; ERGUN, Irem; OKOH, Ifeoma; ALAAGIB, Aisha; MUDANNAYAKE, Oshan; ALYAFEAI, Zaid; CHIEN, Vu; RUDER, Sebastian; GUTHIKONDA, Surya; ALGHAMDI, Emad; GEHRMANN, Sebastian; MUENNIGHOFF, Niklas; BARTOLO, Max; KREUTZER, Julia; ÜSTÜN, Ahmet; FADAEE, Marzieh; HOOKER, Sara (2024). «Aya Dataset: An open-access collection for multilingual instruction tuning». En: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics* [en línea]. Vol. 1. Bangkok: Association for Computational Linguistics, pp. 11521-11567. <https://aclanthology.org/2024.acl-long.620/>. [Consulta: 04/03/2025].

VASWANI, Ashish; SHAZEER, Noam M.; PARMAR, Niki; USZKOREIT, Jakob; JONES, Llion; GOMEZ, Aidan N.; KAISER, Lukasz; POLOSUKHIN, Illia (2017). «Attention is all you Need». *arXiv:1706.03762* [en línea]. <https://doi.org/10.48550/arXiv.1706.03762>. [Consulta: 04/03/2025].

WANG, Yuxia; LI, Haonan; HAN, Xudong; NAKOV, Preslav; BALDWIN, Timothy (2024). «Do-not-answer: Evaluating safeguards in LLMs». *Findings of the Association for Computational Linguistics: EACL 2024* [en línea]. St. Julian's, Malta: Association for Computational Linguistics, pp. 896-911. <https://aclanthology.org/2024.findings-eacl.61/>. [Consulta: 04/03/2025].

WEI, Jason; TAY, Yi; BOMMASANI, Rishi; RAFFEL, Colin; ZOPH, Barret; BORGEAUD, Sebastian; YOGATAMA, Dani; BOSMA, Maarten; ZHOU, Denny; METZLER, Donald. (2022). «Emergent abilities of large language models». *arXiv:2206.07682*. <https://doi.org/10.48550/arXiv.2206.07682>.[Consulta: 04/03/2025].

WOLFRAM, Stephen (2023). «What Is ChatGPT doing ... and why does it work?». *Stephen Wolfram Writings* [en línea]. <https://writings.stephenwolfram.com/2023/02/what-is-chatgpt-doing-and-why-does-it-work>.

YUE, Xiang; QU, Xingwei; ZHANG, Ge; FU, Yao; HUANG, Wenhao; SUN, Huan; SU, Yu; CHEN, Wenhu (2023). «MAmmoTH: Building math generalist models through hybrid instruction tuning». *arXiv:2309.05653* [en línea]. <https://doi.org/10.48550/arXiv.2309.05653>. [Consulta: 04/03/2025].

ZHANG, Shengyu; DONG, Linfeng; LI, Xiaoya; ZHANG, Sen; SUN, Xiaofei; WANG, Shuhe; LI, Jiwei; HU, Runyi; ZHANG, Tianwei; WU, Fei; WANG, Guoyin (2024). «Instruction tuning for large language models: A survey». *arXiv:2308.10792* [en línea]. <https://doi.org/10.48550/arXiv.2308.10792>. [Consulta: 04/03/2025].

Notas

1. Hay cierta variación en la traducción al castellano de *large language models*. Preferimos *modelos grandes de lenguaje* para hacer énfasis en su tamaño.
2. En el anexo se pueden ver algunos ejemplos.
3. Huggingface es una empresa norteamericana que dispone de una plataforma para compartir datos y programas abiertos. La captura es de *https://huggingface.co/datasets/tasksource/jigsaw_toxicity*.
4. Disponible en *https://ai.google.com/research/NaturalQuestions*.
5. Según los autores, un 93 % de textos en inglés, y con algunos textos en otras lenguas.
6. Por ejemplo: *https://www.kaggle.com/datasets/muhammadzain010/arithmaticdata/data*.
7. Por ejemplo: *https://github.com/google-deepmind/mathematics_dataset*.
8. Por ejemplo: *https://huggingface.co/datasets/Vezora/Tested-143k-Python-Alpaca*.

Anexo

Estos son algunos ejemplos de cómo se describe el funcionamiento de ChatGPT en la prensa española.

- «Pero ¿cómo funciona? No da una respuesta automatizada, sino que es capaz de interpretar lo que se le pide», en *https://elpais.com/expres/2024-07-15/miniguia-para-entender-como-funciona-chatgpt.html*.
- «¿Cuáles son las ventajas, si las hay, de ChatGPT frente a motores de búsqueda como Google? Interacción natural: ChatGPT puede comprender y responder a preguntas en lenguaje natural, lo que significa que puedes interactuar con él», en *https://www.computing.es/mundo-digital/opinion/1138491046601/innovacion-de-chatgpt-frente-google.1.html*.
- «Las IA deciden entonces qué contar (y qué omitir), y a partir de estas descripciones los investigadores pudieron definir esas opiniones en preferencias ideológicas», en *https://elpais.com/expres/2024-11-28/chatgpt-de-derechas-gemini-de-izquierdas-por-que-las-ia-no-son-neutrales.html*.
- «Los usuarios han encontrado rápidamente varios casos en los que ChatGPT inventa hechos, personas, datos, y los mezcla en textos muy bien redactados y convincentes», en *https://www.newtral.es/chatgpt-veracidad-inteligencia-artificial-desinformacion/20221215/*.
- «El nuevo modelo de ChatGPT es capaz de percibir emociones, bromear y mostrar distintos estilos y tonos en su voz leyendo un cuento», en *https://www.cuatro.com/noticias/internacional/20240514/nuevo-modelo-chatgpt-interpreta-emociones-inteligencia-artificial_18_012489806.html*.
- «Como decíamos, su mayor diferenciación es la interpretación de la pregunta para adaptar la respuesta, por lo que su capacidad para comprender el contexto y la intención detrás de las preguntas o consultas de los usuarios lo convierten en una herramienta muy útil por la precisión en los sistemas de búsqueda de información», en *https://www.laregion.es/opinion/que-es-que-consiste-famoso-chatgpt_1_20230326-3154655.html*.