Praxis Filosófica
ISSN (I): 0120-4688 / ISSN (D): 2389-9387
DOI: 10.25100/pfilosofica.v0i62S.15097

LOS MODELOS EXTENSOS DE LENGUAJE (LLM) COMO CALCULADORAS DE PALABRAS, UNA PRECISIÓN CONCEPTUAL

Large Language Models (LLMs) as Word Calculators: A Conceptual Clarification

Jorge Francisco Maldonado Serrano
Andrés Felipe Cadena Zambrano

Universidad industrial de Santander, Bucaramanga, Colombia.

Universidad industrial de Santander, Bucaramanga, Colombia.

Resumen

Este artículo propone una conceptualización filosófica y técnica de los Modelos de Lenguaje Extensos (LLM), entendidos como calculadoras estadístico-probabilísticas de palabras. La propuesta se construye a partir de una analogía crítica con la calculadora matemática. La primera sección expondrá la estructuración y el funcionamiento de los LLM desde la analogía de la calculadora. En primera instancia, recorreremos los límites de la analogía, con lo cual buscamos acentuar el potencial de los LLM; en segunda instancia, expondremos el tipo de soporte matemático, como lo es el algebra lineal, de los LLM; se presentará su el funcionamiento estadístico básico del LLM por transformadores. La segunda sección lo específico de los LLM como calculadora de palabras para exponer su diferencia con las bases de datos, el fenómeno de la alucinación y presentar, a manera de conclusión, las posibilidades en el campo de la escritura.

Palabras clave: *LLM; álgebra lineal; epistemología de la inteligencia artificial; redes neuronales; arquitectura de transformador.*

¿Cómo citar?: Maldonado Serrano, J. F. y Cadena Zambrano, A. F. (2025). Los Modelos Extensos de Lenguaje (LLM) como calculadoras de palabras, una precisión conceptual. *Praxis Filosófica,* (62S), e20515097. https://doi.org/10.25100/pfilosofica.v0i62S.15097

Recibido: 07 de julio de 2025. Aprobado: 30 de octubre de 2025.

Large Language Models (LLMs) as Word Calculators: A Conceptual Clarification

Jorge Francisco Maldonado Serrano[1]

Universidad industrial de Santander, Bucaramanga, Colombia.

Andrés Felipe Cadena Zambrano[2]

Universidad industrial de Santander, Bucaramanga, Colombia.

Abstract

This article proposes a philosophical and technical conceptualization of Large Language Models (LLMs), understood as statistical-probabilistic word calculators. The proposal is built upon a critical analogy with the mathematical calculator. The first section presents the structure and functioning of LLMs. First, we explore the limits of the comparison to highlight the potential of LLMs. Second, we explain the mathematical foundations of LLMs, specifically linear algebra. Finally, we outline their basic statistical operations through transformer architectures. The second section focuses on what is specific to LLMs as word calculators, distinguishing them from databases, analyzing the phenomenon of hallucination, and concluding with the possibilities that LLMs open for writing.

Keywords: *LLM; Linear Algebra; Epistemology of AI; Neural Networks; Transformer architecture.*

---

[1] Profesor titular, Escuela de Filosofía, Universidad Industrial de Santander. Licenciado en Filosofía y Letras de la Universidad Santo Tomás, Especialista en Docencia Universitaria de la Universidad El Bosque, Magíster en Filosofía de la Universidad Javeriana, Doctor en Filosofía de la Universidad Autónoma de Madrid. Trabaja en el área de la filosofía de la tecnología, la filosofía del arte y filosofía experimental. Dirige la línea de investigación Lenguajes Digitales. Director del Grupo de Investigación Tiempo Cero.

[2] Profesor asistente Escuela de Derecho y Ciencia Política, Universidad Industrial de Santander. Abogado de la Universidad Autónoma de Bucaramanga, especializado en Derecho Procesal Civil y Derecho Comercial de la Universidad Externado de Colombia. Trabaja en el área de Filosofía de la tecnología, renovaciones tecnológicas en la rama judicial colombiana. Estudia la maestría en filosofía en la Universidad Industrial de Santander, miembro del grupo Tiempo Cero.

LOS MODELOS EXTENSOS DE LENGUAJE (LLM) COMO CALCULADORAS DE PALABRAS, UNA PRECISIÓN CONCEPTUAL

Jorge Francisco Maldonado Serrano

Universidad industrial de Santander, Bucaramanga, Colombia.

Andrés Felipe Cadena Zambrano

Universidad industrial de Santander, Bucaramanga, Colombia.

"Si el cálculo nos aparece como una actividad maquinal, entonces la máquina es el ser humano que realiza el cálculo." (1942, III-20/ IV-20)

I. Introducción

Las reflexiones de Wittgenstein en las *Observaciones sobre los fundamentos de las matemáticas* podrían servirnos perfectamente para entender los grandes modelos de lenguaje (LLM) y aclarar cómo usarlos para desplegar todo su potencial. Se trata de máquinas que calculan palabras, de las cuales no tenemos un manual de uso como lo tenemos de otras máquinas que se han producido; pero se puede decir que calculan sólo en la medida en que haya un ser humano que realice el cálculo con estas. Sin perder de vista esta idea, pero sin hacer ni una exégesis ni una aplicación de la filosofía de Wittgenstein, creemos crucial profundizar en la conceptualización de la inteligencia artificial generativa de lenguaje, realizada en los LLM a partir de la idea de calculadora.

¿Por qué habría de interesarnos esta precisión conceptual? ¿Tiene que ver con su uso? ¿Por qué no tenemos un manual de uso como lo tenemos de otras máquinas? ¿Por qué pensar los LLM como máquinas? Y ¿por qué, precisamente como calculadoras de palabras, cuando estamos acostumbrados a suponer que las calculadoras son sólo de números mientras que los LLM son de lenguaje? Con este artículo pretendemos resolver esta última pregunta.

El interés de explicar por qué los LLM son máquinas que calculan palabras surge de la preocupación que el sistema educativo general (desde la

educación básica hasta la universitaria a nivel de posgrados), especialmente profesoras y directivas, por tener claro cómo afrontar la IA, específicamente los LLM, dado que les resuelven las tareas y ejercicios a estudiantes, desde los problemas matemáticos sencillos hasta escritos elaborados sobre cualquier temática (ensayos, respuestas, trabajos). ¿Qué hacer? ¿Regular externamente los LLM o implementar en las prácticas pedagógicas los LLM?

En términos de regulación se busca establecer un marco ético para el uso de la inteligencia artificial que regule el uso y, por tanto, que permita establecer un marco jurídico que garantice a aquel. Un marco ético de uso significa establecer reglas de utilización de la IA a nivel docente y estudiantil. El marco jurídico despliega los rangos de sanciones respecto de los extremos de las acciones prohibidas por el marco ético, tanto para el productor y comercializador de la IA, como para la usuaria. En este ámbito se ha avanzado en general con directrices para la IA a nivel mundial, todavía no específicamente para la educación, pero constituye ya un marco legal de referencia (IMCO – LIBE, 2024; Acuerdo PCSJA24-12243, 2024; Corte Constitucional República de Colombia, 2024; Corte Suprema de Justicia, 2023; Benites Ocampo, 2023; Comisión Europea para la eficiencia de la justicia, 2018; Stankovich, 2023).

En términos pedagógicos las preocupaciones se reparten entre crear aplicativos para mejorar la formación del estudiante (Choi *et al*., 2023; Meyer *et al.*, 2023); hacer uso directo de los LLM a través de una capacitación un poco acelerada a estudiantes y profesores para integrarla activa y drásticamente en la práctica pedagógica (Kasneci *et al.*, 2023); y seguir con la práctica pedagógica a su ritmo normal para evitar el estrés causado por la tecnología (González Amarilla y Pérez Vargas, 2019).

Sólo el tiempo lo dirá con certeza, pero nos parece que mientras no se aplique una ontología del software (Maldonado Serrano *et al*., 2020) — porque la IA es sólo un software— para entender los LLM, no podremos desarrollar adecuadamente ni un marco ético ni un marco legal idóneo ni una práctica pedagógica. Sin embargo, podemos partir de la analogía con la calculadora para aclarar las posibilidades de los LLM[3]. Y así, sólo después de las suficientes aclaraciones sobre su funcionamiento, será más sencillo discutir marcos éticos y pedagógicos.

En este sentido, proponemos entender los LLM como calculadoras de palabras con el fin de establecer los alcances y límites que tienen. Con ello, evitamos las concepciones o ideaciones distorsionadas de los LLM, que nos

[3] Popova (Popova *et al.*, 2021) ha adelantado una propuesta de una ontología de la IA, pero no consideramos que tengan una comprensión precisa de la IA porque suponen que la IA debe comprenderse desde una comparación con la inteligencia humana.

parece adecuado llamar narrativas ilusionantes porque pueden ser de corte optimista (tecno-optimismos o tecnofilias) o pesimista (tecno-pesimismos o ludismo). Si los LLM son un nuevo tipo de calculadora, entonces ya tenemos una idea, aunque vaga, de su verdadero potencial.

La primera sección expondrá la estructuración y el funcionamiento de los LLM desde la analogía de la calculadora. En primera instancia, recorreremos los límites de la analogía, con lo cual buscamos acentuar el potencial de los LLM; en segunda instancia, expondremos el tipo de soporte matemático, como lo es el algebra lineal, de los LLM; se presentará su el funcionamiento estadístico básico del LLM por transformadores. La segunda sección lo específico de los LLM como calculadora de palabras para exponer su diferencia con las bases de datos, el fenómeno de la alucinación y presentar, a manera de conclusión, las posibilidades en el campo de la escritura.

II. Las redes neuronales y la operación con partes-palabra

Esta analogía de la calculadora de palabras puede resultar muy limitada porque el programa de una calculadora no resulta de un proceso de redes neuronales y porque la calculadora se limita a operar según diez dígitos (de cero a nueve) a 4 bits (modelos avanzados podían llegar a 16 bits), mientras que los LLM operan con más de 150.000 tokens (lo equivalente a los dígitos de la calculadora) a 32 bits y hasta 64 bits[4].

II.1 Límites de la analogía

Podemos poner en suspenso la diferencia de la capacidad computacional de un LLM frente a una calculadora y concentrarnos sólo en la diferencia respecto del programa. Con esto, antes que demostrar la imposibilidad de la analogía, más bien precisamos el hecho de que el LLM puede ser entendido como un tipo de calculadora mucho más versátil que la calculadora matemática.

La calculadora matemática implica un programa lógico-matemático (resultado de un proceso simbólico determinista) que opera soluciones a cálculos matemáticos según introduzca el usuario a través del teclado

[4] Existen procedimientos post-entrenamiento que se denominan *quantization*. Estos procedimientos reducen el número de bits iniciales de los tokens y sus cálculos, con el fin de hacer más manejable o pequeño un modelo para ordenadores con menos potencia computacional. 4K significa que el modelo está cuantizado o reducido a 4 bits, 8k que está cuantizado a 8 bits. Si bien este procedimiento reduce la precisión de un modelo los resultados son aceptables.

numérico y los botones de operación. Sin entrar en calculadoras programables, sinteticemos dos rasgos de este aspecto diferencial:

a. Las teclas tienen valores en sí mismos o anteriores a la entrada que da el usuario: las teclas de número que habilita al usuario a postular uno o dos números, las teclas de operación que lo habilita para que se hagan operaciones entre los números o al número, y las teclas de borrado (simple o total).
b. Las operaciones pueden ser verificadas por el ser humano (diríamos a mano).

La red neuronal implica un programa que, si bien en el fondo es lógico-matemático, opera soluciones de relación entre partes-palabra, según el *prompt* dado por el usuario con todo el teclado alfanumérico. En pocas palabras, el usuario usa el lenguaje natural, no sólo números, para hacer uso del LLM (darle su entrada, *input* o, como se dice técnicamente, un *prompt*).

a. Las teclas no tienen valor alguno, sólo en la medida en que se introduzca el *prompt; la parte-palabra adquiere un valor numérico (como veremos, vectorial); es decir, el valor* se asigna después de la entrada hecha por el usuario.
b. Las operaciones no pueden ser realmente verificadas por el ser humano (podría ser que idealmente lo fuesen, pero realmente es imposible)

No perdamos de vista, pues, que hay una diferencia grande. Con todo, esta diferencia nos abre camino para precisar que la calculadora de palabras, algo antes impensable, ahora no sólo es real, sino que hace cálculos muy certeros, aunque no precisos. Como se trata de cálculos de probabilidad, millones en número, la verificación del cálculo por parte de un ser humano no sólo es improbable y realmente inviable sino innecesaria.

Además, y por último, la calculadora fue construida por circuitos lógico-aritméticos; el LLM requiere de un entrenamiento donde se fijan pesos a sus fórmulas de regularización estadística, que son las que clasifican todo el *dataset* (Textos) que se utilizan para entrenarlo. Aquí encontramos la diferencia entre el paradigma en construcción de máquinas predecibles en su funcionamiento como puede serlo una calculadora o cualquier software programado según un árbol de decisiones o diagrama de flujo predeterminado, y el paradigma de construcción de redes neuronales, donde la solución al problema que debe resolver la máquina (o software para el caso), no se diseña como si se pudiera determinar, sino que es la misma red neuronal la que lo encuentra a manera de probabilidad, aspecto que se detallará un poco más en II.3.

II.2 Vectores o partes-palabra en vez de números

La base numérica de los LLM es el álgebra lineal, una rama de las matemáticas. Ahora bien, cualquier computador es, de suyo, una súper calculadora numérica programable. Tanto así que su gran utilidad y versatilidad radican en poder ayudar en cualquier área de las matemáticas, como una calculadora especializada. Esto se debe a que en computación, como su principio fundamental, cualquier operación matemática se puede llevar a una operación simple de números enteros, como lo estableció Turing en 1936 en su texto '*On computable numbers with an Application to the Entscheidungsproblem*' (Turing, 2004). Por lo tanto, sobra decir que un LLM, por ser un software, en el fondo equivale a una gran calculadora numérica, como cualquier máquina de Turing, pero no será este el eje de nuestro argumento.

En el álgebra lineal, el vector es el objeto de análisis. La idea de vector fue posible gracias la creación de la geometría analítica desarrollada por René Descartes, específicamente a su creación del plano de coordenadas, más conocido como plano cartesiano, que consiste en el cruce de dos rectas numéricas perpendiculares entre sí, llamadas abscisas (eje X horizontal) y ordenadas (Y línea vertical); su punto de cruce se asume como punto cero, cero (0,0) u origen. Por lo que debemos partir de la idea de que el vector existe siempre en un plano y tener presente que el vector más simple existe en un plano de dos dimensiones. Por ello, se ubica o identifica con dos puntos en el plano que se suele escribir así: (x,y).

En realidad, cualquier línea con magnitud o módulo, dirección, sentido y punto de aplicación es un vector en el plano, pero para efectos de nuestra comprensión de la calculadora de palabras sólo nos referiremos a los vectores cuyo origen es el punto (0,0). Así, en un plano de dos dimensiones los vectores se identifican con una pareja de puntos y se nombran con una letra con una flecha sobre ellos: $\vec{x}$. En este sentido, en razón a que las líneas de los ejes son rectas que tienden al infinito, podemos decir que, virtualmente, hay infinitos vectores en un plano.

La importancia de los vectores ubicados en el plano radica en que con ellos se pueden precisar las relaciones entre los mismos vectores. El álgebra lineal sería como la ciencia de las relaciones de vectores. Se relacionan los ejes para conformar el plano, se relacionan los puntos de cada uno de los ejes para identificar los vectores (x,y) y se relacionan los vectores entre sí, por ejemplo dos vectores $\vec{A}$(a,b), $\vec{B}$(c,d), para conocer su proximidad, diferencia o relación. La cercanía se calcula por medio de una operación matemática tan simple como puede serlo una suma. En general, las relaciones

se pueden establecer porque cada vector es único, diferente e inconfundible en su plano, sólo hay un único vector para cada punto en el plano, cuando se trata de planos de tipo euclidiano, esto es, rectos o lisos.

Ahora bien, el álgebra lineal es interesante porque estudia las reglas matemáticas de relación no sólo entre vectores de planos bidimensionales sino las relaciones de vectores de n puntos que corresponden a sus planos de *n* dimensiones: relaciones de suma, resta, multiplicación, división, etc., entre vectores de *n* dimensiones. Planos de más de cuatro dimensiones se suelen denominar hiperplanos, así como vectores de más de cuatro puntos se podrían llamar hipervectores, aunque no se denominan así en álgebra lineal.

Ahora podemos entender lo más importante para el funcionamiento de los LLM. Todos los LLM tienen un conjunto de tokens o vectores, que equivalen a partes de palabra. No son palabras completas, sino partes-palabra porque surgen de un análisis estadístico inicial de los textos, durante su entrenamiento. Lo crucial radica en que debemos concebir la idea de que un token (o vector) de un LLM hoy día suele equivaler, algebraicamente hablando, a un hipervector en hasta 12.289 dimensiones, lo que implica ese mismo número de valores por vector, pesos o parámetros. Estos tokens base de los LLM equivalen a su vocabulario inicial. De ahí que se hable de modelos de, por ejemplo, 4×10^9 parámetros, 175×10^9, y hasta $1{,}5 \times 10^{12}$ parámetros. De los LLM que se han conocido públicamente, se sabe que tienen un número de tokens que oscila entre 32.000 y 100.000. Para ser un poco más precisos, se trata de una matriz de tokens (partes-palabra) con la cual se establecen relaciones entre los tokens; una matriz de 12.289 x 256.000, por ejemplo. Dicho en lenguaje convencional, una tabla de 256.000 vectores, cada uno de 12.289 dimensiones. En algunos modelos, los tokens son frases simples u oraciones que se pueden tomar como una unidad, de modo que, aunque usemos la expresión partes-palabra, asumimos que estos modos de clasificación del lenguaje también se incluyen.

Estos datos varían de modelo a modelo, pero para nuestros fines no nos interesa precisar la potencia computacional de cada modelo, sino de entender que las palabras de los LLM realmente son pedazos de caracteres o frases, que a veces pueden tomarse como palabras u oraciones, y que las relaciones entre las palabras se calculan como se calculan las relaciones entre los vectores: se trata del mismo principio. Por tanto, un LLM, traduce a sus tokens (su vocabulario) el *prompt* inicial que un usuario le da para poder calcular las relaciones que puede establecer. Pero ¿qué relaciones establece? Es lo que necesitamos aclarar a continuación

II.3 Capas de redes neuronales o campos semánticos

Una vez precisada la base relacional fundante de los LLM, proponemos ascender y entender las relaciones superiores que se computan sobre esta base o la forma en que las capas de la red neuronal que conforma un LLM calculan un resultado probabilístico. Este cálculo implica comparaciones de diferentes resultados entre las diferentes capas para escoger, normalmente de forma aleatoria (estocástica), una respuesta definitiva al *prompt*.

Los programas de los LLM son como conjuntos de capas de neuronales que requieren un entrenamiento, mientras que la calculadora matemática es un programa predeterminado que usa la misma lógica computacional para dar las respuestas solicitadas. Aquí está la base de la diferencia señalada como b. en III.1.

Podemos decir que un programa clásico se caracteriza por ser un constructo (una máquina) que se hace para solucionar un problema[5]. Hasta antes de las redes neuronales, el programador debía especificar paso a paso la manera en que el ordenador debía calcular la solución al problema que se buscaba solucionar. En el caso de la calculadora, hay n operaciones o cálculos que son finitos, cuya manera de solucionarlas se establece de antemano. En el caso de un LLM, por cuanto se basa en redes neuronales, la manera de solucionar el problema para el cual se construye la red no está determinada por el programador humano. Justo la característica de la red neuronal consiste en que la manera de llegar a la solución la calcula o establece la red neuronal durante su etapa de entrenamiento, identificando patrones a través de una clasificación de los datos del *dataset*, que se hace, normalmente mediante optimización estadística de funciones de pérdida usando algoritmos como *backpropagation* (retro-propagación) y descenso de gradiente, algo parecido a las regresiones estadística. El entrenamiento de una red neuronal tiene, pues, una sola labor: clasificar los datos en función de la probabilidad de ocurrencia dado su contexto. Esta clasificación o identificación de patrones se logra gracias a los millones de cálculos estadísticos que perfilan la información o *dataset* con el cual se entrena. Dicho en otras palabras, el hecho de que la red neuronal se construya con sistemas estadístico-probabilísticos, garantiza que encuentre la manera de solucionar problemas de forma autónoma, pero por patrones, y no por instrucciones previamente programadas en la red. Por ello, un LLM no requiere de un inimaginable conjunto de reglas (ortográficas, semánticas, sintácticas, etc.) para saber

[5] Cualquier historia de la computación muestra el tipo de problemas de cálculo, sobre todo militares, que debían resolverse a gran velocidad para tener una ventaja competitiva (Ceruzzi, 2003; Haigh y Ceruzzi, 2021).

cómo responder al usuario. ¿Pero cuál es el problema que soluciona las redes neuronales de un LLM, y este en su conjunto?

El problema fundamental de todo LLM es el siguiente: ¿cómo se relacionan las palabras entre sí? Lo que implica clasificar todas las formas y niveles de relación entre las partes-palabra. Como para cualquier red neuronal, se recurre a la estadística y al cálculo de probabilidad. Así, el entrenamiento de un LLM establece los valores de cada token (por ejemplo, 12.288) que se fijan comparando estadísticamente millones, hasta miles de millones de textos. Estos textos digitales (que para algunos modelos son la internet completa) sirven de *dataset* inicial para asegurar esos valores fijos o pesos. Esto quiere decir que para cada token se establece una relación numérica o matemática con todos los demás, de modo que se puede siempre calcular la probabilidad de secuencia entre los tokens.

No viene al caso explicitar el detalle de los cálculos para determinar esta relación, principalmente porque los mismos creadores de los LLM no saben cómo la máquina llega a establecer dichas relaciones. Esto en la medida en que ningún ser humano puede entender realmente cómo cientos de miles de vectores de decenas de miles de puntos se relacionan entre sí. Pero tampoco lo explicitamos porque nuestro interés es filosófico, no computacional. Nótese, sin embargo, que clasificar datos haciendo uso de regresiones estadísticas, no da como resultados reglas o leyes, sólo patrones de comparación[6].

Así, una red neuronal entrenada de un LLM calcula la probabilidad de que una palabra preceda, suceda a otras. Esta cadena de palabras es el resultado de millones de cálculos comparativos que son directamente proporcionales al número de parámetros del modelo; entre más parámetros tenga el modelo mayor capacidad cálculo puede hacer y, con ello, más puede simular el lenguaje natural. Pero no perdamos el norte en este recorrido: un LLM hace cálculos de proximidad o, dicho en otras palabras, relaciones de relaciones entre números que equivalen a partes-palabra. Pero ¿cómo logra un LLM que estas relaciones se presenten como si tuvieran sentido o significado?

Además del papel de los tokens y de las redes neuronales en los LLM, la tecnología revolucionaria es el *transformer* o transformador cuya concepción se estableció en 2016 en el paradigmático documento de Google *Attention is all you need* (Vaswani *et al.*, 2017). El problema fundamental que resuelve

[6] Una discusión importante para una teoría filosófica o lingüística del significado sería si cualquier contenido semántico depende de patrones iterativos que el cerebro encuentra, como sugiere Hofstadter (1980), de manera que las redes neuronales, efectivamente funcionan como funciona el cerebro.

un LLM consiste en que, cuando entra en funcionamiento, como en el chat de ChatGPT, por ejemplo, se puede formular: ¿cuál es la palabra que sigue al *prompt* dado por el usuario? Así, dado un *prompt* (una secuencia de palabras) y convertido a los tokens del LLM, se calcula, la siguiente palabra que más probablemente podría seguir al *prompt*. Por este funcionamiento se puede decir que un LLM predice la siguiente palabra. Si se siguieran prediciendo palabras sin más, tendríamos sólo un funcionamiento de chat-bot. Pero si bien es correcto lo antedicho, la tecnológica de transformador, básica del LLM, supera el resultado calculado por el chat-bot. Miremos de qué se trata.

La respuesta de un LLM consiste en una cadena que se construye con una especie de bucle iterativo en el que siempre se tiene en cuenta el resultado previo. Es decir, si se calcula una nueva palabra que debe seguir a un *prompt* dado, entonces para la siguiente palabra que debe predecirse, se calcula partiendo del *prompt* más la palabra calculada anteriormente (y así sucesivamente). Esto quiere decir que se reinicia el cálculo con el *prompt* modificado o aumentado con el cálculo anterior, pero se sigue aumentando hasta que se calcula que el resultado es satisfactorio.

Esta descripción puede parecer un poco extraña y simple, pero en razón al hecho de que el transformador hace que el cálculo del LLM tenga en cuenta el resultado previo, esta operación se ha llamado autoatención. Es decir, el LLM no sólo tiene en cuenta el prompt para calcular su respuesta, sino que tiene en cuenta cada una de las palabras que va produciendo como cálculo (y, por lo tanto, las oraciones que se completan). Pero realmente, significa que el cálculo se reinicia cientos de veces, de pronto hasta miles de veces (si se producen miles de palabras, con un nuevo *prompt*, aumentado por el mismo LLM en sus resultados previos).

Por último, y sumado a los tres aspectos anteriores, se debe tener presente que un LLM, como red neuronal, opera como si tuviera diferentes grupos de redes que hacen cálculos de relación o proximidad diferentes y que se van complementando entre sí. En efecto, podemos decir que un *prompt* se analiza desde varias perspectivas (secuencia de letras, secuencias de palabras, oraciones, longitud de palabras), que se llaman clústeres de palabras; por ello se conciben *como si* fueran campos semánticos o agrupamientos semánticos, aunque en realidad no hay nada de semántica en estos cálculos, sólo cálculos estadísticos. Pero lo interesante es que la combinación de diferentes tipos de proximidad permite al LLM calcular coincidencias a un nivel superior.

Un ejemplo simple: se le dice al *prompt* que reescriba una rima al estilo de Shakespeare; la palabra 'Shakespeare' delimita un clúster y, por lo tanto, tokens próximos diferentes a si se le dijera estilo Haiku. Por ello, cuanto

más extenso el *prompt* y más preciso o con más palabras específicas, el LLM calcula con más variedad de cercanías y da respuestas más completas.

Por ello, ante una pregunta simple, todos los LLM, bien sean de pocos billones de parámetros hasta trillones, responden más o menos lo mismo. Pero cuando un *prompt* es complejo e incluye muchas precisiones, la respuesta calculada por un LLM más grande (con más parámetros) puede ser más detallada y extensa que la de un LLM más pequeño. Un LLM grande cuenta con capas más densas y con mayor número de capas que uno chico.

La importancia de esta descripción nos lleva a comprender mejor por qué un LLM puede entenderse como una calculadora de palabras y, más precisamente, como una calculadora estadístico-probabilística de palabras. Sinteticemos: La calculadora está programada previamente en el modo de encontrar soluciones, mientras que la red neuronal aprende cómo llegar a una solución por sí misma. En cierto sentido, puede decirse que los LLM se dan sus propias reglas de solución, que en realidad equivale a aprender por patrones.

III. La calculadora estadístico-probabilística de palabras

Una red neuronal es un tipo de inteligencia artificial (otros tipos, como sistemas expertos y algoritmos de búsqueda, no requieren entrenamiento; las redes bayesianas y los árboles de decisiones, junto a otros tipos, requieren entrenamiento) que requiere un entrenamiento. Si ponemos en suspenso la etapa de entrenamiento (y todas sus implicaciones epistemológicas y ontológicas) y nos concentramos en la operación de los LLM, podemos ver su potencia como calculadora estadística de palabras.

El lector habrá notado que no proponemos una respuesta a la pregunta: ¿qué es un LLM? Sino que tenemos como eje de esta exposición: ¿cómo funciona un LLM? De entrada, la respuesta es que funciona como una calculadora estadística de palabras, pero ¿qué quiere decir esto?

En general, cualquier IA de redes neuronales predice o es una máquina predictiva (Agrawal *et al.*, 2022). Cuando se trata de palabras, las predicciones se calculan con millones de operaciones, que contienen miles de millones de parámetros, en un proceso de autoatención. Dicho de manera escueta: un LLM continúa el *prompt* que se le dio inicialmente; esto es lo único que hace, si bien lo hace con múltiples capas de cálculo.

Pero, esta predicción de palabras, por cuanto no incluye realmente ningún contenido semántico de ninguna de las palabras (puesto que todas se han convertido en números), es una predicción lingüística aproximada. Esto quiere decir que un LLM hace cálculos precisos, pero su resultado no

es exacto[7], esto es, hace estimaciones probables, pero no exactas en sentido determinista. Eso significa que los cálculos, en tanto que cálculos, no son para nada creativo, ingeniosos o novedosos. Esto se pierde de vista cuando no se desconoce algún aspecto del funcionamiento del LLM.

Así mismo, podemos ser radicales y afirmar que ningún LLM es una base datos. Muchos consideran que es una nueva forma de encontrar datos o información. Efectivamente, un LLM se entrena con un dataset, que puede equivaler a una base de datos. Y estos suelen ser miles de millones de textos, lo que equivaldría a miles de millones de datos de una especie de base de datos. Sin embargo, la operación durante el entrenamiento sólo se limita a ajustar pesos (los parámetros) en ecuaciones estadístico-probabilísticas.

Un LLM, después del entrenamiento, nunca busca información nueva en bases de datos. De hecho, no hay información del dataset, sólo pesos ajustados o calibrados, no la información como tal. Por eso no sorprende que el juez Vince Girdhari Chhabria haya preliminarmente reconocido que GPT no infringía las leyes de propiedad intelectual, porque no se accedía a ningún texto original cuando se usaba el chat y porque la información como tal no estaba incluida en el modelo: sólo se habían ajustado pesos de ecuaciones.

Es cierto que un LLM puede parecer que da información porque las palabras que tienen un uso técnico y preciso en el *dataset* se integran con esta precisión en los cálculos comparativos del entrenamiento. Así, las respuestas de un LLM pueden coincidir con la realidad. Es más, puede tratarse de datos o información verídica. Pero no olvidemos que su estructuración técnica no consiste en guardar información precisa sino en calcular las relaciones entre palabras. Por cierto, el cálculo de partes-palabra es tan complejo y los *datasets* del entrenamiento tan amplios, con lo cual los pesos quedan muy finamente calibrados, que las respuestas pueden devolver información precisa. En razón de este alto grado de precisión, se puede pensar que las redes neuronales, entrenadas con libros, pueden ser mejores buscadores de información que las bases de datos académicas, con lo cual se resuelven muchos retos técnicos que empezaron hace miles de años (Flanders, 2020). Pero esto es un efecto no esperado de un LLM[8].

[7] Puede hacerse la salvedad de que el último cálculo estocástico (seleccionar al azar de todas las palabras probables a continuación una cualquiera) sea nulo, y siempre se dé el mismo resultado. Pero esto es sólo una estrategia computacional que da la sensación de variación y creatividad al usuario. No hay tal.

[8] Otro análisis filosófico se debe adelantar para profundizar en esta cuestión epistemológica: ¿hay un nuevo orden de conocimiento con las redes neuronales? ¿implica esto que nuestras teorías del conocimiento, de la gramática, del lenguaje, de la literatura, de la argumentación, deben reevaluarse con estos cálculos internos de un LLM?

Realmente un LLM tiene mucho de caja negra porque no se sabe con exactitud cómo logra los resultados esperados. Aunque sí sabemos que el proceso desde la primera década del siglo hasta el 2021, implicó mucho trabajo humano posterior para corregir los constantes errores de un LLM, por ello, Crawford (2021) argumenta que la IA "no es ni inteligente ni artificial" (p. 8) y especialmente narra cómo los humanos siguen interviniendo permanentemente en los modelos LLM para que funcionen bien (Crawford, 2021, p. 63ss). Es decir, realmente la fase de entrenamiento no es tan automática ni de sólo autoaprendizaje, sino que depende de muchos ajustes constantes, de la mano con trabajo humano directo en la etapa de fine-tunning.

No debe sorprendernos, por tanto, que un LLM pueda caer fácilmente en las llamadas alucinaciones. Como no es una base de datos y, aunque puede dar información cierta, su funcionamiento no se estructura para recabar información confiable (verificada) de manera directa sino para encontrar relaciones entre palabras; a nuestros ojos y cerebro semántico, claramente puede alucinar, es decir, cometer errores como inventarse nombres, referencias, hechos, etc. Las respuestas de un LLM equivalen a relaciones propuestas, que pueden contener información no verídica, sólo probable. Por ello, si se necesita precisión, cualquier asistencia de la IA debe ser rigurosamente verificada.

III.1 ¿Cómo usar una calculadora de palabras?

Concebir un LLM como una entidad pensante equivale a concebir la calculadora matemática como si fuera un matemático cuando sólo se compone de circuitos. Las calculadoras funcionan, en sus diferentes versiones, como máquinas que parecen calcular, pero no calculan, puesto que mueven engranajes o hacen circular energía por circuitos de cierta manera y rotan cintas con números o encienden pequeñas luces (LED) (en los años sesenta desarrollan funciones científicas). Así, nos parece claro que una IA de lenguaje, en tanto que calculadora de palabras, no llegará a tener más conciencia que una calculadora matemática (Larson, 2022), ni a tomar decisiones más allá de las decisiones de los usuarios que la utilicen y de las de los diseñadores de las mismas redes neuronales. Claro, encuentra patrones por sí misma, o aprende, y resuelve a su manera el problema fundamental para el cual fue diseñada: proseguir el *prompt*. Pero esto no implica que tenga agencia, por lo menos no en un sentido humano[9]. Y aunque

[9] Esta es otra área de posible debate que no pretendemos desarrollar en este artículo, pero podría tenerse en cuenta la discusión de la agencia en los objetos técnicos, a sabiendas

da la impresión de que, porque escribimos expresiones como 'encuentra por sí misma', gana un puesto como sujeto, realmente sólo se trata de una limitación de nuestro lenguaje.

El diseño o código puesto en engranajes o transistores físicos es decidido por el ser humano (programador). En igual medida, la decisión de qué operación hacer también la hace el humano (usuario) restringido por las capacidades de la calculadora en cuestión, claro está. Así como la calculadora no se da a sí misma la operación a resolver, un LLM no se da a sí mismo su *prompt* y, por tanto, no decide calcular palabras de manera independiente de una decisión humana. Entonces, ¿cómo usar un LLM?

La regla de oro, según lo que hemos planteado, consiste en lo siguiente: acotar los clústeres que se requieren para una tarea específica. Una de las grandes virtudes descubiertas sobre los LLM es su gran precisión para elaborar código: se requiere que el *prompt* describa con mucha precisión y suficiente detalle la rutina que se quiere hacer, y el resultado puede sorprender. Por lo mismo y tanto, cuando se trata de un asunto de escritura, la mejor estrategia siempre es dar un *prompt* lo suficientemente rico en palabras que le permitan al LLM delimitar su cálculo y el resultado de manera que el resultado resulte más benéfico.

No olvidemos que, al igual que una calculadora matemática que no entiende de números ni del problema que detrás de los números puede haber (Tenía X dinero, gasté Y, ¿cuánto me queda?; X-Y=Z), la calculadora de palabras tampoco entiende ni una palabra de lo que se le escribe. Con todo, así como Z es la respuesta exacta de una calculadora matemática, un LLM da una respuesta precisa al *prompt*. Recuerde, siempre es en función del *prompt*, puesto que el LLM no exige ni da más rigor a este (no lo mejora, sólo lo amplía extensionalmente), y devuelve un resultado son la precisión que el *prompt* le habilita.

Pero ya queda claro que su uso es interesante y ahora es cuando la calculadora de palabras se muestra en su potencial. Si se delimita a unos textos específicos, un LLM puede ser preciso en los análisis de dichos textos, porque se le está dando de antemano el límite de usos. Realmente queremos decir que en muchas áreas donde se usa el lenguaje, tener una novedosísima calculadora de palabras puede acelerar los procesos de trabajo, en proporción directa a la capacidad del usuario para acotar los textos en los cuales se necesita. En otras palabras, no puede haber un instructivo de uso de un LLM, sino que debe haber un proceso de entrenamiento adicional por parte del usuario.

de que todo software, como el LLM es una máquina (Moreno Ortiz, 2020, pp. 17-82).

Actualmente se han diseñado aplicativos académicos (GPT Scholar, Notebook LLM de Google, Scopus IA) que combinan búsquedas de información en bases de datos académicas con preguntas académicas en reemplazo de las ecuaciones de búsqueda bibliométricas limitadas a 'OR', 'AND', 'NOT' y otros buscadores de proximidad o variaciones en terminaciones. En estos casos, la búsqueda de información sería más "conceptual" porque la LLM puede, por la clasificación de palabras en clústeres semánticos, calcular mejor según lo que calcule como significado del *prompt*. Sin olvidar que tenemos software diseñado, no debería extrañarnos que haya usos diferenciales entre los usuarios que pagan por el servicio y los que no (similar al acceso a bases de datos).

Para terminar esta parte, debe quedar claro que la fortaleza más grande que tienen los LLM es su capacidad de re-escritura. Antes habíamos mencionado que el *prompt* inicial se debe convertir a tokens. Esto implica que hay una especie de re-organización de este a partir del cual se construye el resto de la respuesta. En este sentido, la potencialidad de un LLM para sintetizar textos, clasificar —lo que para nosotros serían— ideas, conceptos, o argumentos es importante, así sea sólo de manera literal. Casi podríamos afirmar que un LLM tiene como función básica volver a escribir lo que se le pida, razón por la cual, traducciones, revisión de redacción, de estilo o de clarificación serían tareas relativamente sencillas. En estos casos, por ejemplo, no se trata de pedirle un ensayo con ideas que pueden ser alucinatorias, sino que revise un escrito y calcule una escritura más plausible.

Pero ¿significa esto que va a primar una especie de minería de textos? Al contrario, nos parece que podemos entender que el LLM puede llevar a un enriquecimiento de la minería de textos al operar sobre estructuras probabilísticas complejas.

III.2 A manera de conclusión

La analogía del LLM como calculadora estadístico-probabilística de palabras muestra que su alcance es de apoyo en tareas precisas, no de creación escritural, así muchos estén tentados a decir que piensa y que es creativa. Se trata de tener una máquina que ayuda con procesos de escritura donde las reglas precisas e iterativas sean la clave. Así como en arquitectura, gracias a los programas de dibujo de planos, eliminaron la tarea de los delineantes de arquitectura; es probable que muchos expertos que trabajen en ciertas áreas de manejo de lenguaje reemplacen, si no su labor, gran parte de sus ejercicios iterativos (redacción, puntuación, claridad, incluso argumentación) (Mirzakhmedova *et al.*, 2024).

Todavía queda mucho terreno por recorrer para poder especializar los LLM en áreas tan complejas como las humanidades, en especial la filosofía. Al lado de software de gestión bibliográfica o de análisis cualitativo (CAQDAS – Computer Assisted Qualitative Data Analysis Software), surge este LLM como máquina que perfecciona búsquedas, marcado de textos y análisis cualitativo. Así, con el recurso de este software que opera con lenguaje natural, cualquier actividad de análisis puede ser optimizada con la riqueza que nos puede dar el lenguaje natural.

Referencias bibliográficas

Acuerdo PCSJA24-12243 [Consejo Superior de la Judicatura]. *Por el cual se adoptan lineamientos para el uso y aprovechamiento respetuoso, responsable, seguro y ético de la inteligencia artificial en la Rama Judicial*. 16 de diciembre de 2024. Recuperado de: https://actosadministrativos.ramajudicial.gov.co/GetFile.ashx?url=%7e%2fApp_Data%2fUpload%2fPCSJA24-12243.pdf

Agrawal, A., Gans, J. y Goldfarb, A. (2022). *Prediction Machines: The Simple Economics of Artificial Intelligence.* Harvard Business Review Press.

Benites Ocampo, C. A. (2023). Detectando el Fraude con Inteligencia Artificial: Una Perspectiva Avanzada en Auditoría Forense. *Revista la Junta*, *6*(2), 13-40. https://doi.org/10.53641/junta.v6i2.116

Ceruzzi, P. E. (2003). *A History of Modern Computing* (2nd ed.). MIT Press.

Choi, E. P. H., Lee, J. J., Ho, M.-H., Kwok, J. Y. Y., y Lok, K. Y. W. (2023). Chatting or cheating? The impacts of ChatGPT and other artificial intelligence language models on nurse education. *Nurse Education Today*, *125*, 105796. https://doi.org/10.1016/j.nedt.2023.105796

Comisión Europea para la eficiencia de la justicia. (3 de diciembre de 2018). *Carta ética europea sobre el uso de la inteligencia artificial en los sistemas judiciales y su entorno*. Recuperado de: https://protecciondata.es/wp-content/uploads/2021/12/Carta-Etica-Europea-sobre-el-uso-de-la-Inteligencia-Artificial-en-los-sistemas-judiciales-y-su-entorno.pdf

Corte Constitucional República de Colombia. (2024). *Sentencia T-323-24*. Recuperado de: https://www.corteconstitucional.gov.co/relatoria/2024/t-323-24.htm

Corte Suprema de Justicia. Sala de Casación Civil, Agraria y Rural. *Sentencia SC370-2023*. M. P. Aroldo Wilson Quiroz Monsalvo. 10 de octubre de 2023. Recuperado de: https://img.lalr.co/cms/2023/10/11204140/Sentencia-sobre-Uber-SC370-2023.pdf

Crawford, K. (2021*). Atlas of AI: Power, Politics, and the Planetary Costs of Artificial Intelligence.* Yale University Press. https://doi.org/10.12987/9780300252392

Flanders, J. (2020). *A Place for Everything: The Curious History of Alphabetical Order*. Pan Macmillan.

González Amarilla, S. B., y Pérez Vargas, S. F. (2019). Tecnoestrés docente: El lado opuesto de la utilización de las nuevas tecnologías por los Docentes del Nivel Medio. *Revista Científica Estudios e Investigaciones*, *8*(1), 21-35. https://doi.org/10.26885/rcei.8.1.21

Haigh, T. y Ceruzzi, P. E. (2021). *A New History of Modern Computing*. MIT Press. https://doi.org/10.7551/mitpress/11436.001.0001

Hofstadter, D. R. (1980). *Gödel, Escher, Bach: An Eternal Golden Braid*. Penguin Books.

IMCO – LIBE. (2024). *Artificial Intelligence Act*. European Parliament.

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok, G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., Weller, J., Kuhn, J. y Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and Individual Differences, 103*, 102274. https://doi.org/10.1016/j.lindif.2023.102274

Larson, E. J. (2022). *El mito de la inteligencia artificial: Por qué las máquinas no pueden pensar como nosotros lo hacemos*. Shackleton Books.

Maldonado Serrano, J. F., Rodríguez Ramírez, D. A., Caceres, P. B., y Petit Suárez, J. F. (2020). An Ontology of Software: Series, Structure and Function. *Praxis Filosófica*, (51), 115-131. https://doi.org/10.25100/pfilosofica.v0i51.10114

Meyer, J. G., Urbanowicz, R. J., Martin, P. C. N., O'Connor, K., Li, R., Peng, P.-C., Bright, T. J., Tatonetti, N., Won, K. J., Gonzalez-Hernandez, G. y Moore, J. H. (2023). ChatGPT and large language models in academia: Opportunities and challenges. *BioData Mining*, *16*(20). https://doi.org/10.1186/s13040-023-00339-9

Mirzakhmedova, N., Gohsen, M., Chang, C. H. y Stein, B. (2024). Are Large Language Models Reliable Argument Quality Annotators? En P. Cimiano, A. Frank, M. Kohlhase y B. Stein (Eds.), *Robust Argumentation Machines. RATIO 2024. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 14638) (pp.129-146). Springer. https://doi.org/10.1007/978-3-031-63536-6_8

Moreno Ortiz, J. C. (Ed.) (2020). *Tecnología, agencia y transhumanismo.* Ediciones USTA. https://doi.org/10.15332/li.lib.2020.00263

Popova, A. V., Balashkina, I. V., Nikitin, P. V. y Prokopovich, G. A. (2021). Ontology of Artificial Intelligence: On the Question of the Dehumanization of Social Regulators. En E. G. Popkova, V. N. Ostrovskaya y A. V. Bogoviz (Eds.), *Socio-economic Systems: Paradigms for the Future* (pp. 81-88). Springer International. https://doi.org/10.1007/978-3-030-56433-9_10

Stankovich, M. (2023). *Kit de herramientas globales sobre IA y el Estado de derecho para el poder judicial*. UNESCO.

Turing, A. (2004). On Computable Numbers, with an Application to the Entscheidungsproblem (1936). En B. J. Copeland (Ed.), *The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life: Plus The Secrets of Enigma* (pp. 58-90). Oxford University Press.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. y Polosukhin, I. (2017). *Attention Is All You Need*. ArXiv. https://doi.org/10.48550/arXiv.1706.03762

Datos de financiación del artículo

Los autores declaran que no recibieron financiación para este artículo.

Implicaciones éticas

Los autores no tienen ningún tipo de implicación ética que se deba declarar en la escritura y publicación de este artículo.

Declaración de conflicto de interés

Los autores declaran que no tienen ningún conflicto de interés en la escritura o publicación de este artículo.

Contribuciones de los autores

Jorge Francisco Maldonado Serrano: escritura (borrador original), escritura (revisión del borrador y revisión/corrección).
Andrés Felipe Cadena Zambrano: escritura (borrador original), escritura (revisión del borrador y revisión/corrección).

Autor de correspondencia

Jorge Francisco Maldonado Serrano. jmaldona@uis.edu.co. Carrera 27 Calle 9 Bucaramanga, Santander.

Declaración de uso de inteligencia artificial

Los autores declaran que no utilizaron ningún programa o aplicación de inteligencia artificial.