El primer software. Breve historia de la Inteligencia Artificial 5

Diego Germán González | 31/01/2023 19:11 |

En los artículos anteriores vimos como la investigación sobre Inteligencia Artificial pasó de la

frivolidad de intentar distinguir a un humano de una máquina o a la imitación de la arquitectura del cerebro a la imitación por software del proceso del pensamiento.

Si los primeros avances vinieron de físicos, biólogos, fisiólogos y matemáticos**, el próximo gran salto vendría de un lugar inesperado, las ciencias políticas.**

Simon y la racionalidad

Si estudiaste Administración de Empresas probablemente hayas tenido que empollarte un grueso libraco llamado *El comportamiento administrativo*. Para lo que suele ser la bibliografía de la carrera es un libro que resulta bastante útil e interesante, aunque un poco denso.

El autor es un señor que **recibiría el premio Nobel de Economía por rebatir uno de los dogmas más queridos dela Ciencia Económica.** El del consumidor racional.

Graduado en Ciencias Políticas **su carrera comenzó estudiando las administraciones municipales** y después de un breve paso por el organismo administrador del Plan Marshall cofundó y enseñó en el postgrado de Administración Industrial de la que ahora se conoce como Universidad Carnegie Mellon.

¿Cuál es el punto en común entre las burocracias y la Inteligencia Artificial? **El proceso de toma de decisiones**.

Los economistas clásicos afirmaron siempre que somos decisores racionales. Es decir que ante una serie de alternativas, empresarios o consumidores vamos a elegir aquella opción que maximice más los beneficios o reduzca más los costos. La conclusión de esto es que ante la misma serie de alternativas y circunstancias todos tomaremos la misma decisión.

Simon minimizó el alcance de esa supuesta racionalidad. Sostuvo que el decisor nunca considera todas las alternativas disponibles y que no todos usamos los mismos criterios a la hora de evaluarlas. Lo que si hacemos es aplicar los mismos criterios a todos los problemas como si fuera una receta de cocina. Eso fue la base del a heurística o programación basada en reglas.

Otro aporte de Simon adoptado por la Inteligencia Artificial **es la división de metas en submetas más pequeñas.** Alcanzar las submetas facilita alcanzar la meta general.

El primer software de Inteligencia Artificial

Con la ayuda de Allen Newell, graduado en Física, y C Shaw, un actuario transformado en programador de computadoras, **Simon comenzó el desarrollo de Logic Theorist, considerado como el primer programa de Inteligencia Artificial de la historia.**

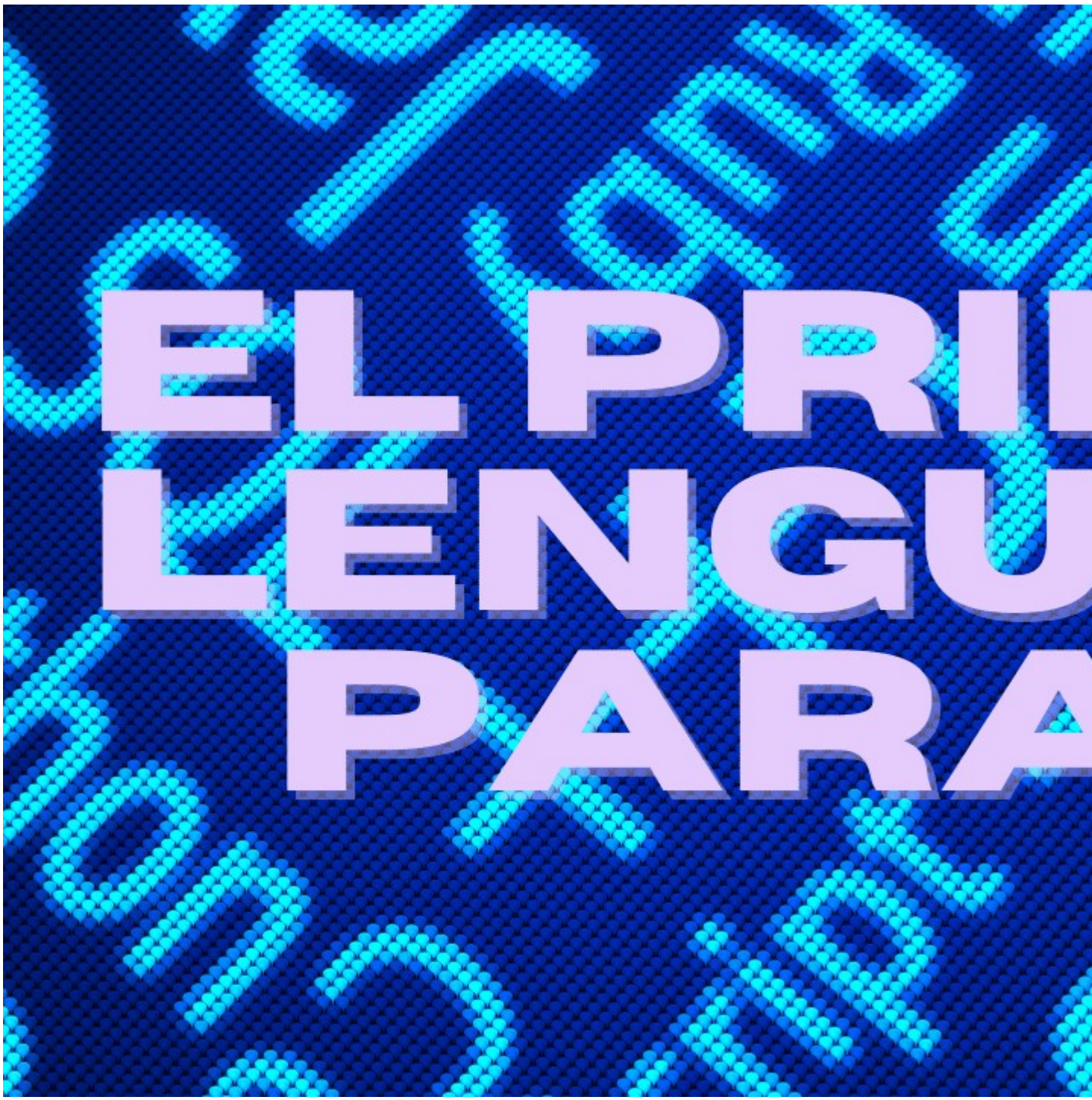
Aunque la intención original era que el programa resolviera problemas de ajedrez o geométricos, finalmente lo destinaron a resolver teoremas de un conocido libro de Matemáticas. Sin embargo**, a diferencia de la máquina de Turing el objetivo no era la resolución de problemas matemáticos sino emular la forma en la que los humanos mediante heurísticas selectivas determinaban el paso siguiente** que debían realizar.

La búsqueda de la respuesta correcta se puede representar en forma gráfica como una estructura en forma de árbol. A este gráfico se lo conoce como árbol de búsqueda.

En la raíz del árbol de búsqueda se halla la hipótesis inicial. De la raíz salen ramas en las que se ubican variaciones de la hipótesis inicial que son el resultado de aplicar sobre esta las reglas de la lógica. A cada una de las ramas se aplican otras manipulaciones generando subramas. El proceso se repite hasta alcanzar la conclusión deseada.

El objetivo del programa de Simon y sus compañeros no era la demostración del teorema sino encontrar el camino que permitiera llegar a esa demostración. La aplicación exploraba el árbol de acuerdo con ciertas reglas prefijadas para encontrar la rama que era más probable que condujera al resultado correcto. Seguía repitiendo el proceso hasta encontrar el camino correcto.

Si los primeros intentos de Inteligencia Artificial fueron por el lado de imitar la arquitectura del cerebro, Simon y sus colegas siguieron el camino contrario. Imitaron el funcionamiento de una computadora con personas. Antes de empezar la tarea de codificación un grupo de estudiantes a los que se unieron la esposa y los hijos de Simon recibieron fichas con las subrutinas y las reglas lógicas expresadas en inglés y simularon el comportamiento de los componentes del programa.



En nuestra entrega anterior contamos como Simon, un teórico de las Ciencias Políticas junto a un físico llamado Newell y a un actuario devenido en programador de apellido Shaw comenzaron la construcción del primer programa de inteligencia artificial conocido como Logical Theorist. Esto requirió la invención **del primer lenguaje de programación específico para la Inteligencia Artificial**

Habíamos dejado esta historia con el trio junto a colaboradores y familia simulando el comportamiento de las diferentes partes del programa usando personas y fichas manuscritas.

Luego de varias simulaciones como estas se implementó el programa en una computadora real. La prueba fue un éxito ya que **el software logró demostrar treinta y ocho teoremas de unos de los capítulos del libro Principia Matemática de Russell y Whitehead**. Incluso en uno de los casos (Y

sin tener instrucciones específicas de hacerlo) encontró una forma de probarlo mucho más «elegante» que la de los autores del libro.

Índice

El primer lenguaje de programación para Inteligencia Artificial

El que Simon y su equipo se tomaran tanto tiempo para escribir su programa es porque **necesitaban de un lenguaje de programación específico que tuviera el suficiente poder y flexibilidad para sus propósitos.** Ese lenguaje se llamó IPL (Por las siglas en inglés de Lenguaje de Procesamiento de Información) e introdujo por primera vez la técnica de procesamiento de listas para programación.

IPL se diferenciaba de los lenguajes de alto nivel de la época en que **no requería que los símbolos se definieran de antemano y que tenía la capacidad de asociar y modificar estructuras de símbolos.**

La llamada técnica de procesamiento de listas consiste en **almacenar cada pieza de información junto con indicaciones sobre cómo encontrar piezas de información asociadas a ellas.** Cambiando las indicaciones se pueden construir nuevas asociaciones.

El «Solucionador General de Problemas»

Para crear su próximo software Simon y Newell decidieron probar un enfoque diferente. En la época circulaba una investigación psicológica que invitaba a los participantes a explicar en voz alta la forma en la que resolvían problemas lógicos. El duo descubrió que esas formas eran completamente diferentes a las que utilizaba su software por lo que decidieron hacer su propia versión de la investigación y **crear un software a partir de los métodos descriptos por los participantes.** El programa (Conocido como GPS por las siglas en inglés para Solucionador General de Problemas) estaba codificado en base a una organización de la información y heurísticas independientes de las tareas que se les pidiera realizar.

Esta nueva metodología recibió el nombre de «Análisis de medios a fines» y consiste en **comparar la situación actual con la ideal y tomar acciones que reduzcan la diferencia entre ellas para luego volver a hacer la evaluación hasta que la diferencia se reduzca a cero**. Esta metodología permite al programa reaccionar ante modificaciones en las variables del problema. El programador indica el problema y una llamada tabla de diferencias en las que se indican los cursos de acción posibles y en qué circunstancias lo son.

GPS era capaz de descomponer un problema en subproblemas y aplicar el enfoque de retroceso, es decir que si un camino no funcionaba retrocedía y seguía por otro.

Durante los 11 años que estuvo en funcionamiento, **GPS resolvió acertijos, realizó integración simbólica y rompió códigos secretos.**

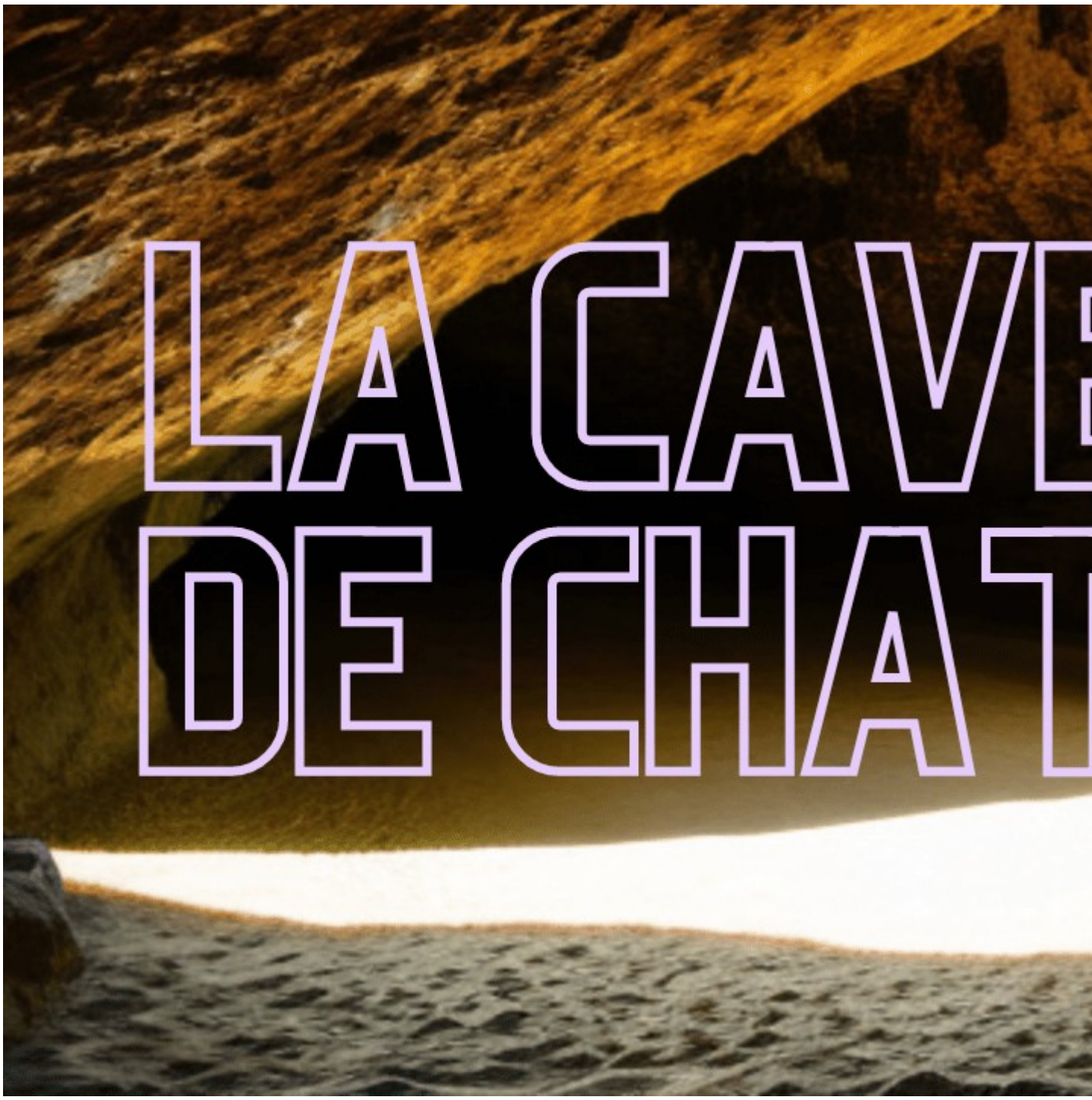
Mientras Simon y Newell se entretenían con esto, un estudiante llamado Robert K. Lindsay desarrolló un programa conocido como SAD SAM. El soft **era capaz de extraer información de sentencias del tipo «Juan es el hijo de Pepa» y «Juan es el hermano de Alberto» y construir un**

árbol genealógico deduciendo que Alberto también es hijo de Pepa (No tengo ni idea de cómo se las arreglaría con las familias ensambladas del mundo actual.

Por supuesto, que el gigante de la industria informática de la época, IBM, no podía quedarse afuera de la investigación sobre inteligencia artificial, un campo que en plena guerra fría ya se revelaba cómo de enorme potencial para aplicaciones militares y, en el próximo artículo hablaremos de sus primeras contribuciones en el campo.

La caverna de ChatGPT

Diego Germán González | 16/02/2023 22:46 |



Dicen que para novedades los clásicos. **Una alegoría escrita cuatro siglos años de nuestra era resulta ideal para entender cuáles son los límites de las nuevas aplicaciones de Inteligencia Artificial.** me refiero a la «Caverna de ChatGPT» que no es ni más ni menos que una adaptación de la famosa alegoría de la caverna de Platón

No tengo nada que objetar al uso de las herramientas de inteligencia artificial. De hecho, encuentro que facilitan mucho el trabajo. Pero, siempre y cuando **sea usada por personas que tengan conocimientos suficientes como para evaluar su trabajo.**

Por poner un ejemplo; uno puede pedirle a ChatGPT que escriba un plugin de WordPress, pero si se carece de conocimientos sobre PHP ese plugin puede producir graves problemas de seguridad.

La alegoría de la caverna

Platón fue un filósofo griego que vivió entre los siglos V y IV antes de Cristo. Expresaba sus pensamientos en la forma de mitos y alegorías. **La más conocida de ellas fue la de la caverna.**

Publicada en *La República*, la alegoría imagina a **un grupo de personas encadenadas en una caverna, detrás de ellos tienen un fuego que arroja sombras sobre la pared que tienen frente a ellos. Las sombras son lo único que ven e imaginan que son lo único que existe ignorando lo que hay más allá.**

Cuando uno de los prisioneros es liberado puede ver el mundo como realmente es y se da cuenta de lo limitado de sus experiencias en la caverna.

Según los estudiosos de Platón, esta alegoría pone de relieve que todos vivimos nuestras vidas basándonos en nuestras propias informaciones y experiencias. Informaciones y experiencias equivalentes a las sombras de la caverna. Al igual que los prisioneros, existe la verdadera realidad y está más allá de nuestra comprensión.

La caverna de ChatGPT

ChatGPT y sus competidores tienen tanto admiradores como detractores. Pero, nadie había dado una explicación técnica sobre sus fallas hasta un artículo publicado en New Yorker por el escritor de ciencia ficción Ted Chang

Para explicar las fallas de los modelos de lenguaje Chang hace una analogía con lo que sucede con imágenes y archivos de audio.

La grabación y reproducción de un archivo digital requiere de dos pasos: el primero es la **codificación, momento en el cual el archivo se convierte a un formato más compacto, y a continuación la decodificación, que es el proceso inverso**. El proceso de conversión se denomina sin pérdida (el archivo restaurado es igual que el original) o con pérdida (Parte de la información se perdió para siempre). La compresión con pérdida se aplica en archivos de imágenes, video o audio y la mayoría de las veces no es perceptible. Cuando lo es se denomina artefacto de compresión». Los artefactos de compresión se presentan en la forma de partes borrosas en imágenes o sonido metálico en audio.

Chang usa la analogía de un JPG borroso de la web para referirse a los modelos de lenguaje. Y, esta es bastante exacta. **Ambos comprimen la información conservando solo «Lo importante». L**os modelos de lenguaje generan, a partir de grandes cantidades de datos de texto, una representación compacta de los patrones y relaciones entre palabras y frases.

A partir de ella se genera texto nuevo tratando en lo posible que sea similar en contenido y significado al texto original. El problema es cuando en la web no hay información suficiente para que se pueda generar un texto nuevo. Esto se traduce en que ChatGPT pueda escribir un ensayo de nivel universitario, pero no hacer sencillas operaciones de 5 dígitos.

Chang concluye que:

> Incluso si es posible restringir que los grandes modelos de lenguaje participen en la creación, ¿deberíamos usarlos para generar contenido web? **Esto tendría sentido solo si nuestro objetivo es reempaquetar la información que ya está disponible en la Web.** Algunas empresas existen para hacer precisamente eso; generalmente las llamamos

> fábricas de contenido. Quizás la borrosidad de los modelos de lenguaje les sea útil, como una forma de evitar la infracción de derechos de autor. Sin embargo, en términos generales, diría que cualquier cosa que sea buena para las fábricas de contenido no es buena para las personas que buscan información. **El aumento de este tipo de reenvasado es lo que nos dificulta encontrar lo que estamos buscando en línea en este momento**; cuanto más se publica en la Web el texto generado por grandes modelos de lenguaje, más se convierte la Web en una versión más borrosa de sí misma.

Y, como los prisioneros de la caverna, nuestra experiencia sería mucho más pequeña de lo que la realidad nos ofrece.