



# Un recorrido por los modelos de lenguaje:

Desde Shannon a GPT-4



## FELIPE BRAVO MÁRQUEZ

Profesor Asistente del Departamento de Ciencias de la Computación de la Universidad de Chile, Investigador Asociado del Centro Nacional de Inteligencia Artificial (CENIA) e Investigador Joven del Instituto Milenio Fundamento de los Datos (IMFD). Realizó su doctorado en el grupo *machine learning* de la Universidad de Waikato, Nueva Zelanda. Sus intereses de investigación y experiencia se centran en la adquisición de conocimientos e información a partir del lenguaje natural, abarcando las áreas del procesamiento del lenguaje natural (NLP), el aprendizaje automático (ML), la inteligencia artificial (AI) y la recuperación de información (IR). En su investigación, ha desarrollado varios métodos de NLP y ML para el análisis de opiniones y emociones en medios de comunicación social, así como otras aplicaciones centradas en la equidad, la salud y la educación, entre otras.

✉ [fbravo@dcc.uchile.cl](mailto:fbravo@dcc.uchile.cl)

✂ [@felipebravom](https://twitter.com/felipebravom)



**RESUMEN.** Los modernos modelos de lenguaje, representados por asistentes virtuales y chatbots como ChatGPT y Google Bard, han transformado la manera en la que nos relacionamos con las máquinas, permitiéndonos interactuar con ellas de la misma forma con la que interactuamos con nuestros pares humanos, usando el lenguaje.

Estas impresionantes capacidades no son el resultado de un mero golpe de suerte, sino el fruto de un progresivo desarrollo basado en descubrimientos científicos e innovaciones tecnológicas en el campo del aprendizaje automático y el procesamiento del lenguaje natural. En este artículo, trazamos el recorrido desde sus inicios, desde los primeros modelos de lenguaje estudiados por Shannon en la década de 1950, pasando por los primeros modelos de lenguaje neuronales propuestos por Bengio y otros, hasta llegar a los actuales grandes modelos de lenguaje.

## Introducción

La llegada de ChatGPT a finales del año 2022, en conjunto con la proliferación de modelos de lenguaje y asistentes de conversación durante este mismo año, ha impactado profundamente en nuestro quehacer diario. Hemos sido testigos de cómo las máquinas han adquirido la capacidad de utilizar algo tan propio a nosotros los humanos: el lenguaje. Este salto tecnológico ha generado debates profundos sobre sus implicaciones en diversos ámbitos, tales como la propiedad intelectual, la diseminación de información falsa, el impacto en la salud mental y la educación, y, en última instancia, su potencial para reemplazar el trabajo humano. En este artículo, no nos adentraremos en dichos debates, sino que nos concen-

## *Se espera que un modelo de lenguaje atribuya probabilidades a las oraciones en función de su coherencia, tanto desde un punto de vista semántico como sintáctico.*

traremos en trazar una línea histórica de los descubrimientos y avances tecnológicos que han permitido a las máquinas alcanzar la habilidad de escribir de manera similar a nosotros.

La capacidad de dominar el lenguaje humano ha estado en el imaginario de la computación desde sus inicios. En 1950, Alan Turing propuso el famoso “Test de Turing”, que planteaba la pregunta de si era posible crear una máquina capaz de mantener una conversación con otra persona sin que ésta pueda distinguir si está conversando con un humano o una máquina. En 1964, Joseph Weizenbaum creó Eliza, uno de los primeros agentes de conversación desarrollado en el MIT. Eliza era un programa que simulaba a un psicoterapeuta y utilizaba reglas predefinidas para responder a las conversaciones del usuario. En paralelo, en la década de 1950, Claude Shannon realizó los primeros estudios sobre cómo modelar el lenguaje escrito de manera estadística y predictiva [1]. Usando técnicas de teoría de la información, calculó la dificultad de predecir palabras en base a las anteriores a partir de un “corpus” de texto. Sin embargo, en 1957, Noam Chomsky, un lingüista y científico cognitivo, cuestionó la capacidad de los modelos estadísticos para capturar la gramática del lenguaje humano [2]. Para ilustrar esto, presentó dos oraciones ficticias:

1. Colorless green ideas sleep furiously.
2. Furiously sleep ideas green colorless.

Aunque ambas oraciones carecen de significado, Chomsky argumentó que sólo la primera se considera gramaticalmente correcta. Sin embargo, un mode-

lo de lenguaje como los estudiados por Shannon consideraría ambas oraciones igualmente poco probables. Las ideas de Chomsky detuvieron el progreso en el modelado estadístico de lenguaje por varios años.

## Primeros modelos de lenguaje

Un modelo de lenguaje es una herramienta que asigna una probabilidad a cada posible oración que se puede formar a partir de un conjunto finito de palabras. Tomemos, por ejemplo, las siguientes oraciones: 1) “El perro ladra” y 2) “Baila tuerca alto”. Se espera que un modelo de lenguaje atribuya probabilidades a las oraciones en función de su coherencia, tanto desde un punto de vista semántico como sintáctico. Por consiguiente, debería asignar una probabilidad mayor a la primera oración en comparación con la segunda.

La motivación original de estos modelos se origina en el problema de reconocimiento del habla o de transcripción automática. Si consideramos las siguientes frases en inglés: “recognize speech” (reconocer el habla) y “wreck a nice beach” (arruinar una playa bonita), notamos que ambas suenan prácticamente idénticas (o, técnicamente hablando, producen la misma señal acústica). Un modelo de lenguaje construido a partir de un corpus de texto debería asignar una probabilidad mayor a la primera frase, dado que es más común y, por lo tanto, se encuentra con mayor frecuencia dentro del corpus. Esto, a su vez, habilitaría al sistema de transcripción para generar el texto correcto. De



hecho, los modelos de lenguaje resultan sumamente útiles en cualquier tarea de procesamiento de lenguaje natural (o NLP por sus siglas en inglés) que involucre la generación de texto (como traducción automática, la generación de resúmenes y los chatbots), ya que ayudan a los sistemas a discernir entre diversas posibles salidas mediante las probabilidades otorgadas por el modelo de lenguaje.

Matemáticamente, el modelo de lenguaje modela una oración como una secuencia de palabras, donde cada palabra es tratada como una variable aleatoria discreta que proviene de un conjunto finito de palabras llamado vocabulario. El modelo tiene la capacidad de asignar probabilidades a cualquier posible oración:

$$p(s) = p(w_1, w_2, \dots, w_n)$$

Técnicamente hablando, esta función es una función de masa de probabilidad multivariada. La regla de la cadena de probabilidades permite descomponer esta probabilidad en un producto de probabilidades condicionales, siguiendo la secuencia de la oración de izquierda a derecha y condicionando la probabilidad de cada palabra en relación a todas las anteriores, a las que llamamos "contexto":

$$p(w_1, w_2, \dots, w_n) = p(w_1) \times p(w_2|w_1) \times p(w_3|w_1, w_2) \times \dots \times p(w_n|w_1, \dots, w_{n-1})$$

Sin embargo, estimar esta función de probabilidad a partir de un corpus de entrenamiento (una colección de texto) se vuelve problemático cuando se condiciona una palabra por una secuencia muy larga de palabras anteriores. Esto se debe a que la cantidad de combinaciones posibles de palabras crece

exponencialmente con la longitud de la secuencia, lo que dificulta tener suficientes instancias en el corpus para estimar las probabilidades con precisión.

Para abordar esta dificultad, se recurre frecuentemente a un enfoque "Markoviano", que implica restringir la memoria de palabras anteriores (o del contexto) dando lugar a los modelos de lenguaje de  $n$ -gramas<sup>1</sup>. Por ejemplo, en un modelo de lenguaje de bigramas, se asume que la probabilidad de una palabra depende únicamente de su predecesora:

$$p(w_3|w_1, w_2) = p(w_3|w_2)$$

Luego, la probabilidad de una oración se simplifica de la siguiente manera:

$$p(w_1, w_2, \dots, w_n) = p(w_1) \times p(w_2|w_1) \times p(w_3|w_2) \times \dots \times p(w_n|w_{n-1})$$

Entonces, el proceso de entrenamiento de un modelo de lenguaje de bigramas se reduce a estimar las probabilidades condicionales de una palabra dada otra palabra, lo que requiere calcular las frecuencias de palabras individuales (unigramas) y de secuencias de dos palabras (bigramas):

$$p(w_i|w_{i-1}) = \frac{\text{count}(w_{i-1}, w_i)}{\text{count}(w_{i-1})}$$

Ejemplo:

$$p(\text{york}|\text{nueva}) = \frac{\text{count}(\text{nueva}, \text{york})}{\text{count}(\text{nueva})}$$

No obstante, los modelos de lenguaje de bigramas o trigramas (que consideran dos palabras anteriores como contexto) tienen limitaciones en contextos largos y no pueden aprovechar contextos similares. Por ejemplo, consideremos los contextos:

c1. Después de comer cereales

c2. Luego de desayunar avena

Aunque esperaríamos que las distribuciones de probabilidad  $p(w|c1)$  y  $p(w|c2)$  fueran similares, dado que c1 y c2 casi no comparten palabras, los modelos de  $n$ -gramas que se limitan a contar frecuencia de palabras no pueden capturar estas similitudes entre contextos.

Una característica importante de los modelos de lenguaje es su capacidad generativa. Pueden crear oraciones nuevas mediante un proceso de muestreo secuencial basado en probabilidades condicionales estimadas. Cada palabra seleccionada se convierte en contexto para elegir la siguiente palabra, replicando la idea de extraer bolas de una urna donde los tamaños de las bolas representan las frecuencias relativas determinadas por el modelo. También es posible partir con un texto inicial a completar y seleccionar consecutivamente la palabra más probable en cada paso, lo que equivale a predecir la siguiente palabra. Esta mirada predictiva cobra mayor relevancia en los modelos de lenguaje neuronales, que serán discutidos a continuación.

---

## Modelos de lenguaje neuronales

---

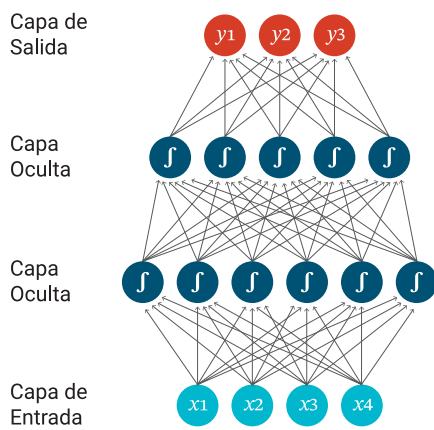
Las redes neuronales [3, 4] son una familia muy popular de modelos de aprendizaje automático, compuestos de unidades de cómputo llamadas neuronas. Cada neurona recibe entradas y salidas escalares, en donde a cada entrada se le asigna un peso escalar denotado como "w". El proceso que sigue una neurona implica la multiplicación de cada entrada por su peso correspondiente, seguida por la suma de estos productos.

---

1 Un  $n$ -grama es una secuencia contigua de  $n$  palabras, cuando  $n=2$  tenemos un bigrama,  $n=3$  un trigramo y así sucesivamente.

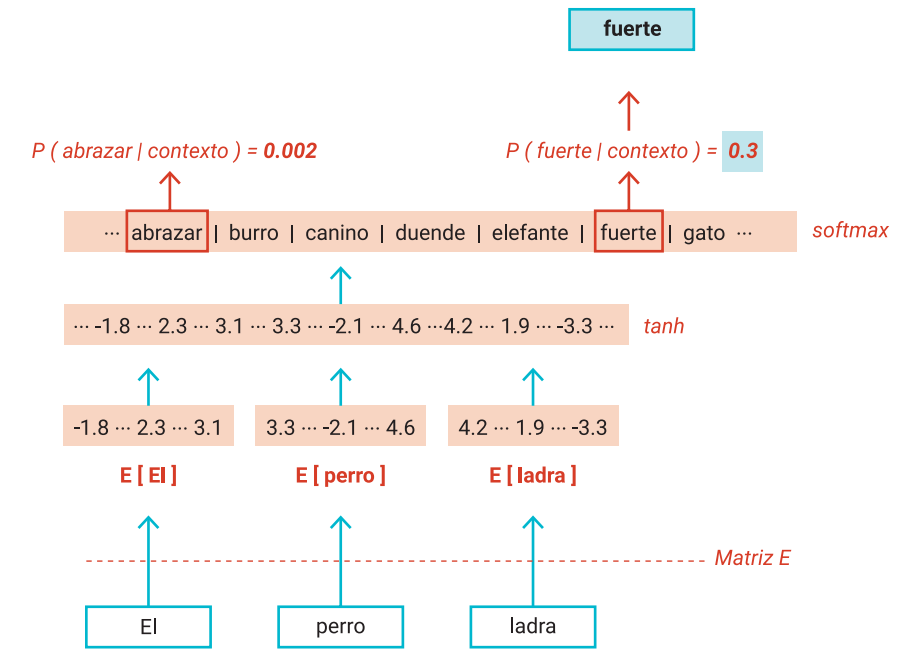


Lo que resulta particularmente fascinante, es que las palabras con significado relacionado (por ejemplo, sinónimos), tienden a adquirir vectores o embeddings cercanos en el espacio vectorial.



**Figura 1.** Ilustración de red neuronal de 4 capas: una entrada, dos capas intermedias u “ocultas” y una capa de salida.

Posteriormente, se aplica una función de activación, generalmente de naturaleza no lineal, al resultado obtenido, el cual se transmite como salida de la neurona. Típicamente, estas neuronas se organizan en capas, las cuales pueden ser apiladas para formar una red neuronal de múltiples capas conocida como “feed-forward”. Las salidas de las capas inferiores se convierten en las entradas de las capas superiores, generando así una progresión de la información a través de la red. La primera capa de la red representa la entrada del modelo, como por ejemplo una imagen o un texto, mientras que la última capa corresponde a la salida deseada, que podría ser una categoría o incluso otro texto. Las capas intermedias suelen llamarse como “capas



**Figura 2.** Modelo de lenguaje neuronal [5]. Las palabras en el contexto se transforman en vectores de dimensionalidad igual a la del vocabulario. Cada palabra se codifica utilizando la técnica *one-hot*, asignando un valor de 1 a la posición correspondiente a la palabra en el vocabulario y 0 a todas las demás posiciones. Estos vectores luego son proyectados en una capa de *embeddings*, donde se convierten en vectores densos. Estos vectores resultantes se combinan en una capa intermedia y posteriormente se proyectan hacia la salida, cuya dimensionalidad coincide con la del vocabulario. Finalmente, se aplica la función *softmax* para generar una distribución de probabilidad que abarca todas las palabras posibles en la salida.

ocultas” y el número de capas determina la profundidad de la red (ver Figura 1).

Los parámetros de una red neuronal (los pesos asociados a todas las neuronas de todas las capas) son inicializados con valores aleatorios, los cuales se ajustan y aprenden a partir de conjuntos de datos etiquetados. Estos consisten en ejemplos compuestos por pares de entradas y salidas deseadas. Este proceso de aprendizaje se lleva a cabo mediante algoritmos de optimización, siendo el algoritmo de Backpropagation el enfoque comúnmente utilizado. Una vez que una red ha sido entrenada de manera adecuada, es capaz de realizar predicciones precisas de las salidas con una alta probabilidad de acierto.

Las redes neuronales han experimentado un vaivén de entusiasmo y escepticismo a lo largo de su historia. Indudablemente, en la última década, han entrado en una fase de popularidad sin precedentes debido a su capacidad para generar “representaciones” de los datos en sus capas intermedias, lo que ha dado lugar a la creación de una disciplina conocida como aprendizaje profundo o *deep learning*. Los hitos significativos en el desarrollo de esta tecnología incluyen el surgimiento de arquitecturas especializadas de redes neuronales, como las redes convolucionales (CNN), las redes recurrentes (RNN) y la arquitectura Transformer. Además, ha sido de suma importancia el empleo de hardware especializado, como las GPUs



(graphic processing units) y TPUs (tensor processing units), para llevar a cabo un entrenamiento eficiente de estas redes, así como el acceso a grandes volúmenes de datos procedentes de la Web y plataformas de Crowdsourcing. Estas últimas permiten llevar a cabo el etiquetado masivo de datos, los cuales se utilizan en el proceso de entrenamiento de las redes neuronales.

En el año 2000, Bengio y colaboradores propusieron utilizar redes neuronales *feed-forward* para construir modelos de lenguaje [5]. Utilizando la regla de la cadena y restringiendo el contexto a un tamaño fijo de palabras (por ejemplo, 5), se puede modelar la probabilidad  $p(w/c)$  mediante una red neuronal. En este enfoque, las palabras anteriores que conforman el contexto se convierten en las entradas de la red neuronal, y la palabra siguiente es la salida deseada. Luego se aplica una función denominada *softmax*, que transforma las salidas de la red (que es un vector de puntajes para cada palabra del vocabulario) en una distribución de probabilidad. De esta forma es posible tomar un corpus de texto y recorrerlo, extrayendo ventanas de palabras junto con sus palabras siguientes para entrenar la red neuronal y desarrollar un modelo de lenguaje (ver Figura 2).

Sin duda, la propiedad más interesante de los modelos de lenguaje neuronales es la noción de *word embedding* o representación vectorial de palabras. Esto implica proyectar las palabras discretas presentes en el contexto hacia vectores densos de cientos de dimensiones (ver Figura 3). Lo que resulta particularmente fascinante, es que las palabras con significado relacionado (por ejemplo sinónimos), tienden a adquirir vectores o *embeddings* cercanos en el espacio vectorial. Este fenómeno encuentra su raíz en una teoría lingüística denominada hipótesis distribucional [6, 7], que sostiene que las palabras presentes en contextos parecidos suelen compartir significados. Como resultado, los modelos de lenguaje neuronales se distinguen de sus contrapartes

### Matriz de embeddings

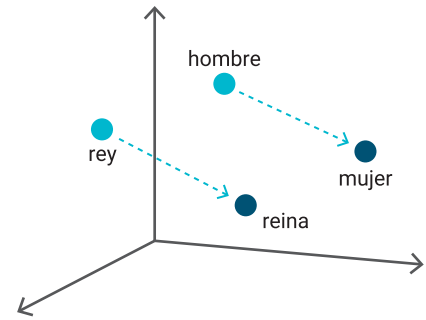
$ V  \times d$				
-6.6	4.3	...	-2.4	← abrazar
3.6	-1.9	...	4.1	← canino
-1.8	2.3	...	3.1	← el
2.5	3.4	...	-2.7	← fuerte
5.0	-3.4	...	1.4	← grueso
4.2	1.9	...	4.2	← ladra
3.3	-2.1	...	4.6	← perro
-2.6	1.5	...	1.6	← zancudo

**Figura 3.** Ilustración de la matriz de *embeddings* aprendida por el modelo de lenguaje neuronal. La matriz consta de filas equivalentes al número de palabras en el vocabulario, con cada fila representada por un vector de dimensión  $d$  (usualmente entre 100 y 500). Cada componente en estos vectores es un valor escalar.

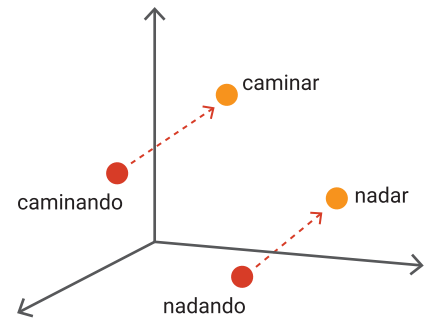
basadas en  $n$ -gramas al ser capaces de aprovechar contextos similares aunque estos difieran en las palabras utilizadas.

Tuvieron que pasar 13 años para que los modelos de lenguaje neuronales fueran adoptados de forma masiva. Esto fue posible gracias al lanzamiento del software de código abierto llamado Word2Vec [8], el cual permitía entrenar vectores de palabras empleando modelos similares a los propuestos por [5], pero con una eficiencia superior. Este software permitió a miles de usuarios entrenar sus propios vectores de palabra utilizando sus colecciones de documentos particulares, lo que les permitió explorar las características intrínsecas de sus palabras. Una propiedad descubierta fue la capacidad de realizar analogías semánticas a través de operaciones aritméticas en el espacio vectorial de las palabras tales como “hombre es a mujer como rey es a reina”, relaciones verbales como “nadar es a nadando como caminar es a caminando”,

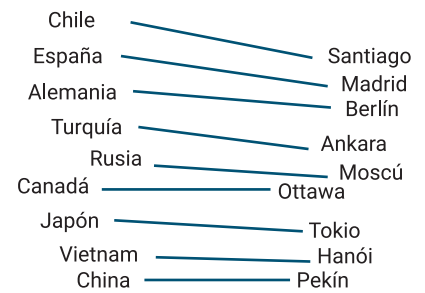
### Masculino / Femenino



### Forma verbal



### País / Capital



**Figura 4.** Visualización de relaciones capturadas por los vectores de palabras entrenados con Word2Vec. Para lograr una representación gráfica, los vectores de alta dimensionalidad se suelen reducir a 2 o 3 dimensiones, permitiendo así su visualización en un sistema de coordenadas.

además de establecer relaciones entre países y sus capitales, como “Santiago es a Chile como Madrid es a España”, entre otras (ver Figura 4).



Aunque los modelos de lenguaje neuronales originales brindan la posibilidad de aprovechar contextos similares a través de vectores de palabras, aún presentan limitaciones en su capacidad para capturar contextos largos. Consideremos el siguiente contexto como ejemplo: “Felipe nació en Chile, donde cursó sus estudios de ingeniería, y posteriormente continuó con su educación de postgrado en Nueva Zelanda. A pesar de que Felipe utiliza el idioma inglés con fluidez, su lengua materna sigue siendo el \_\_\_\_\_”.

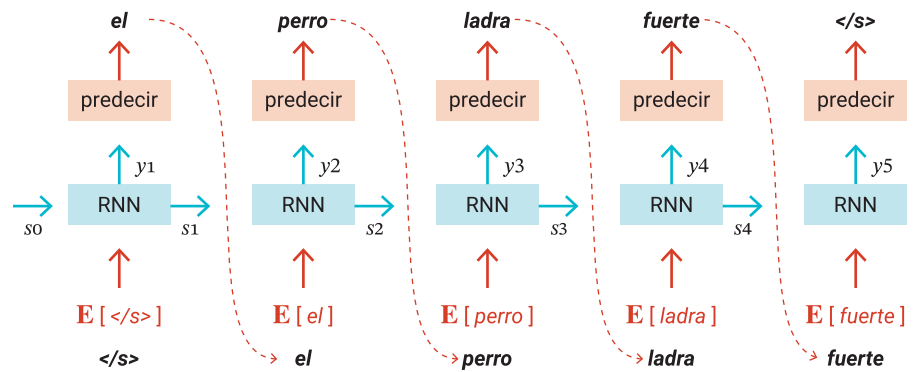
Para lograr predecir con precisión la palabra siguiente, ya sea “español” o “castellano,” un modelo de lenguaje debería tener la capacidad de procesar un contexto lo suficientemente largo para considerar el hecho que Felipe nació en Chile, lo cual es caro computacionalmente para el modelo de lenguaje neuronal original.

Con el objetivo de potenciar a los modelos de lenguaje neuronales con la habilidad de procesar contextos extensos, se han empleado redes neuronales recurrentes [9]. Estas redes, a diferencia de las tradicionales redes *feed-forward*, tienen la capacidad de procesar secuencias de largo variable. En términos generales, cuentan con un vector que representa su estado, el cual se va actualizando a medida que se avanza en la lectura de nuevas palabras reteniendo el historial completo de la secuencia, lo que posibilita la modelación de secuencias de mayor longitud. Finalmente, este estado es utilizado para predecir la palabra siguiente con mayor precisión que el modelo neuronal original (ver Figura 5).

## Los modelos de lenguaje son multitarea

Una propiedad muy poderosa de los modelos de lenguaje neuronales es que no necesitan texto etiquetado para ser

**[GPT-3] tuvo un impacto revolucionario en el ámbito del aprendizaje automático y NLP, ya que introdujo la noción de contar con un modelo único para abordar múltiples tareas, evitando el costoso proceso de entrenar modelos individuales.**



**Figura 5.** Modelo de lenguaje usando una red neuronal recurrente. Cada palabra en la secuencia se convierte en un vector que se procesa de manera secuencial para actualizar el vector de estado  $s$ . Se generan predicciones en cada uno de estos estados.

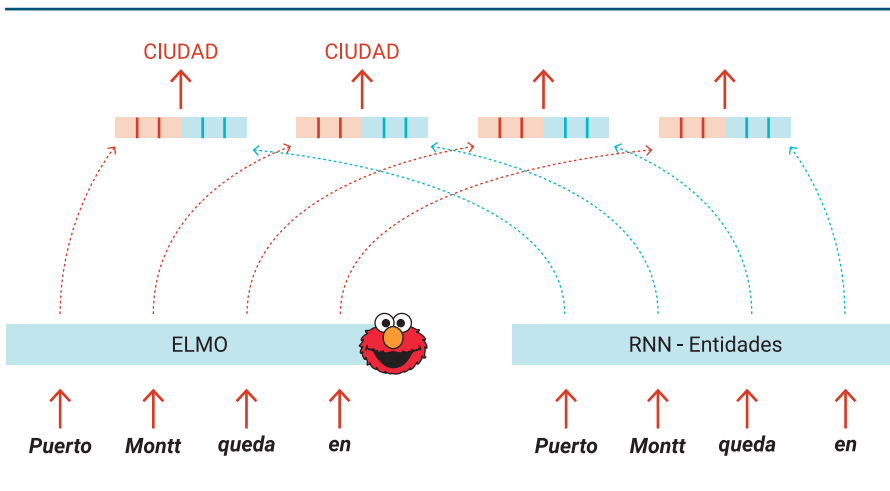
entrenados. En NLP, cada tipo de tarea específica, como la traducción automática, la clasificación de documentos según sentimiento, la generación de resúmenes, o la generación de respuestas a una pregunta, se resuelve entrenando redes neuronales sobre texto manualmente etiquetado con las respuestas esperadas. Etiquetar estos datos es un proceso sumamente costoso, pues requiere intervención de etiquetadores humanos. Por lo general, los datasets de entrenamiento para estas tareas no suelen tener más de miles de ejemplos, lo cual es una limitante para tener modelos que realmente sirvan (o generalicen) para datos distintos a los usados en el entrenamiento.

Si pensamos en la tarea de predecir palabras que ataca el modelo de lenguaje, podemos apreciar cómo este desafío fuerza a la red neuronal a aprender información de índole sintáctica, semántica y de conocimiento general.

Consideremos los siguientes tres contextos:

1. Abróchate el cordón de tus ...
2. Este regalo es para mi ...
3. Cristóbal Colón descubrió ...

Para predecir con alta confianza las palabras “zapatos” o “zapatillas” en el primer contexto, el modelo de lenguaje debe reconocer que estos objetos llevan cordones. Luego, en el segundo contexto, para predecir términos como “novio”, “amigo” o “mamá”, la red neuronal debe entender que un sustantivo (o una frase nominal) es la forma correcta de completar oración, lo que implica un conocimiento gramatical. Finalmente, en el tercer caso, para predecir “América”, el modelo debe poseer la información de que Cristóbal Colón fue el descubridor de América, lo que implica tener conocimiento general.



**Figura 6.** Ejemplo de cómo usar las representaciones de ELMO para resolver la tarea de detectar entidades (ej: ciudades) en una oración. En este ejemplo se tiene una red recurrente para resolver el problema y los estados de ésta son concatenados con los vectores obtenidos con ELMO.

Viendo el modelo de lenguaje de esta forma, hace sentido entrenar un modelo de lenguaje sobre un corpus grande (Wikipedia, Libros, la Web) y aprovechar todo el conocimiento adquirido para resolver tareas más específicas (traducir, clasificar) donde los datos etiquetados son de mucho menor tamaño que el corpus usado para entrenar el modelo de lenguaje.

Esta fue la idea que exploraron Peters *et al.* en 2018 para desarrollar ELMO [10]. ELMO es un modelo de lenguaje compuesto por aproximadamente 100 millones de parámetros y entrenado mediante una red neuronal recurrente, utilizando un corpus de texto grande. La propuesta central de ELMO consiste en aprovechar las representaciones proporcionadas por el modelo de lenguaje previamente entrenado, con el fin de enriquecer redes neuronales más específicas diseñadas para abordar tareas como la traducción o la clasificación, utilizando conjuntos de datos de entrenamiento relativamente pequeños. El proceso implica la pasada inicial de la entrada por el modelo de lenguaje preentrenado, empleando luego los estados de la red recurrente como vecto-

res de palabras contextualizados. Estos vectores son posteriormente suministrados a la red que aborda la tarea específica, lo que potencia la precisión en la resolución de dicha tarea (ver Figura 6).

Para comprender esta idea con mayor claridad, consideremos las siguientes dos oraciones:

1. Me senté en el banco a esperarte.
2. La fila del banco está muy larga.

Supongamos que nuestra tarea consiste en traducir estas oraciones del español al inglés. Una red neuronal común emplearía inicialmente una capa de *embedding*, donde cada palabra se asignaría a un vector. Sin embargo, surge un problema aquí: tanto la palabra “banco” en la primera oración como en la segunda recibirían el mismo vector, a pesar de que sabemos que poseen significados distintos debido a la situación de polisemia. No obstante, al introducir primero las oraciones en un modelo de lenguaje preentrenado, los estados de la red recurrente contextualizan los vectores, generando así vectores distintos para las dos apariciones de “banco”.

Esto permite una traducción más precisa al inglés (“bench” para la primera oración y “bank” para la segunda). De hecho, ELMO mostró mejoras significativas en varias tareas de NLP respecto al estado del arte de su momento, al aprovechar el conocimiento adquirido por el modelo de lenguaje.

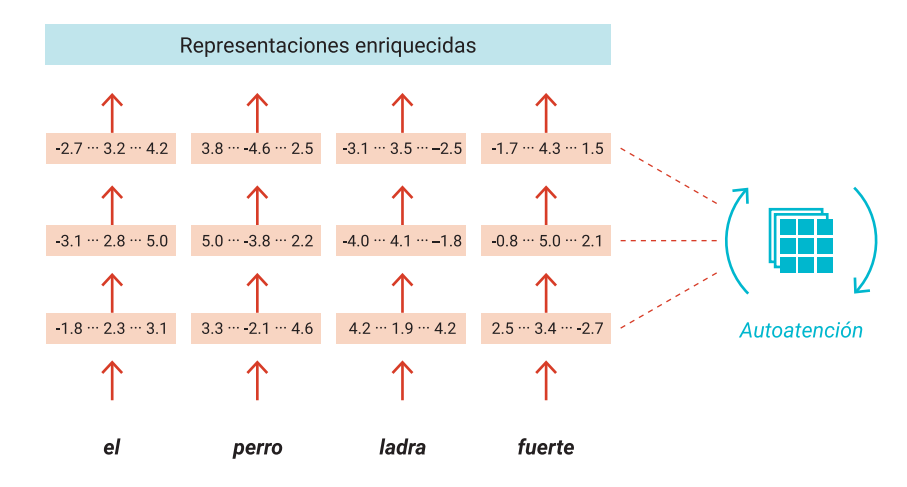
ELMO estableció un precedente en la noción de que un modelo de lenguaje no sólo se encarga de asignar probabilidades a las oraciones, sino que también nos permite adquirir representaciones a partir de grandes colecciones de texto. Estas representaciones posteriormente pueden ser empleadas para resolver tareas específicas de NLP (a las cuales en inglés se les denomina *down-stream tasks*). Sin embargo, los modelos basados en redes neuronales recurrentes (RNN) como ELMO presentan una limitación crucial cuando se trata de expandir su capacidad a través de una mayor cantidad de parámetros. Las RNN requieren procesar la entrada de manera secuencial, lo que restringe la posibilidad de paralelizar sus cálculos. Sin paralelización efectiva no es factible escalar los modelos a los miles de millones de parámetros que tienen los modelos del estado del arte para entrenarlos en un tiempo razonable.

Para enfrentar estos desafíos, en 2017, Vaswani *et al.* introdujeron el Transformer, un tipo de red neuronal basada en mecanismos de (auto)atención [11]. De una manera simplificada, el Transformer recibe una oración representada como una secuencia de vectores, para luego mediante ponderaciones y multiplicaciones de matrices, ir contextualizando estos vectores (o sea, que estos cambien en función de sus vecinos), un procedimiento que puede ser repetido en varias ocasiones para producir una secuencia de vectores altamente enriquecidos y contextualizados (ver Figura 7). A diferencia de las redes neuronales recurrentes (RNN), los Transformers no están limitados por la necesidad de procesar las

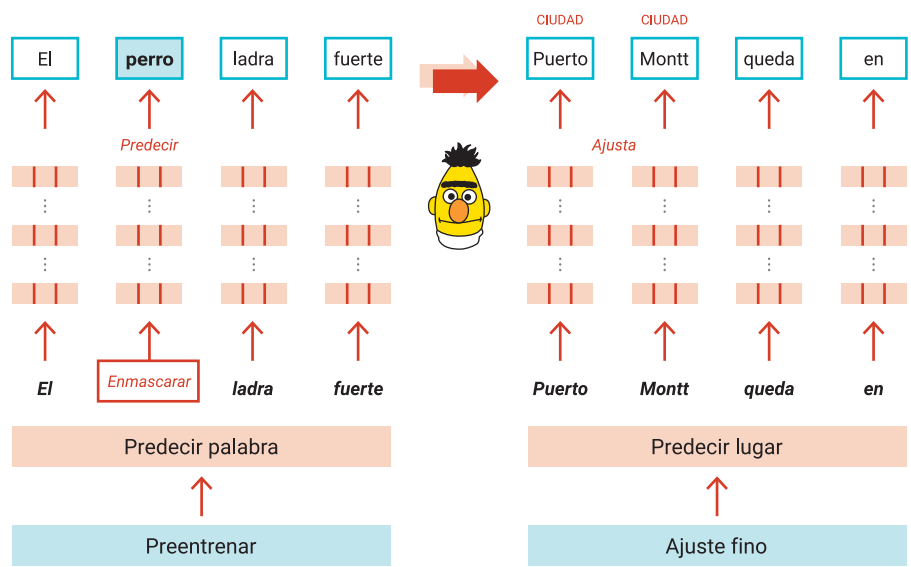
entradas de forma secuencial. Esto los hace particularmente adecuados para la ejecución en hardware especializado como GPUs y TPUs, lo que, a su vez, los convierte en altamente paralelizables.

En 2019, surgió BERT (Bidirectional Encoder Representations from Transformers), que replicó la idea de ELMO pero utilizando un Transformer en lugar de una RNN [12]. BERT tenía 335 millones de parámetros y utilizaba la autoatención del Transformer para reemplazar la recursión de las RNN. Esto permitió obtener *embeddings* contextualizados de manera más escalable en términos de paralelismo y cantidad de parámetros y datos. A diferencia de ELMO, donde estos vectores contextualizados se introducían directamente en la red neuronal de la tarea específica, BERT introdujo la noción de "ajuste fino" o *fine-tuning*. Aquí, el concepto consiste en aprovechar los pesos aprendidos por BERT a partir del corpus de texto grande y focalizarse únicamente en adaptar la última capa de la red para que sea compatible con la tarea en cuestión (por ejemplo, mientras que para el modelo de lenguaje las salidas posibles son todo el vocabulario, en un problema de clasificación por sentimientos sólo se tienen categorías positivo, negativo y neutral). A continuación, la red completa se entrena con los datos etiquetados de la tarea objetivo, con la particularidad de que los pesos, en lugar de ser inicializados de manera aleatoria, parten con la configuración aprendida previamente por BERT (ver Figura 8). Este enfoque obtuvo resultados aún superiores a ELMO.

De manera paralela a BERT, la compañía OpenAI presentó una serie de modelos de lenguaje también basados en la arquitectura Transformer, denominados Generative Pretrained Transformer (GPT). Esta serie incluye GPT-1, GPT-2 y GPT-3, cada uno con una dimensión de parámetros mayor que su predecesor. No obstante, lo que más destaca en estos modelos son las propiedades



**Figura 7.** Diagrama de la arquitectura del Transformer. Se aplican varias capas de atención multicabeza donde los vectores de la entrada se multiplican por distintas matrices y se ponderan entre sí hasta obtener representaciones enriquecidas.



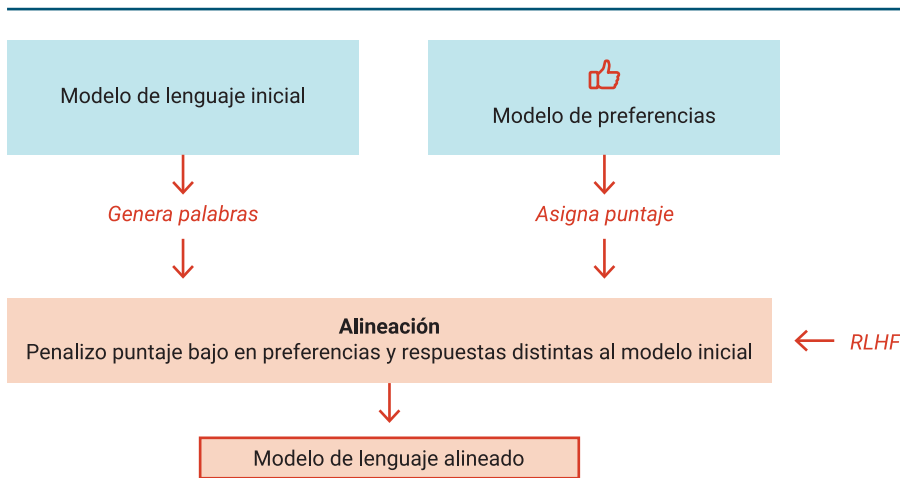
**Figura 8.** Ilustración del proceso de *fine-tuning* con BERT. El modelo original se entrena con la tarea de modelo de lenguaje enmascarado que consiste en predecir palabras faltantes de una oración. Luego el proceso de *fine-tuning* consiste en adaptar la capa de salida según la tarea objetivo y continuar ajustando los pesos de la red con los datos de la tarea objetivo.

“emergentes” que se manifestaron al aumentar el tamaño de los parámetros. El modelo GPT-2 [13] introdujo la noción de que un modelo de lenguaje grande puede ser visto como un modelo multitarea no supervisado, es decir, capaz

de abordar diversas tareas sin requerir el proceso de *fine-tuning* con datos etiquetados como se hacía con BERT.

En términos generales, se puede argumentar que prácticamente cualquier





Instrucción:

"Traduce esta palabra al inglés:  
manzana →"

Ejemplos:

- zanahoria → carrot
- limón → lemon
- lechuga → lettuce
- pera → pear"

Este descubrimiento tuvo un impacto revolucionario en el ámbito del aprendizaje automático y NLP, ya que introdujo la noción de contar con un modelo único para abordar múltiples tareas, evitando así el costoso proceso de entrenar modelos individuales y etiquetar grandes volúmenes de datos. Es importante considerar que las habilidades emergentes sólo se manifiestan cuando escalamos los modelos a los miles de millones de parámetros.

**Figura 9.** Ilustración del Proceso RLHF para alinear las respuestas del modelo de lenguaje. Se tiene un modelo de lenguaje inicial y un modelo de preferencias entrenado con etiquetadores humanos capaz de asignarle un puntaje a las salidas del modelo de lenguaje. El modelo resultante del ajuste fino se aprende optimizando una función que penaliza respuestas de puntaje bajo según el modelo de preferencias como también las que se alejan demasiado del modelo original.

tarea de NLP puede ser "codificada" como una instrucción dentro del contexto del modelo de lenguaje, para que luego la propiedad generativa del modelo de lenguaje le permita resolver la tarea en cuestión. Por ejemplo, sería factible usar como contexto la instrucción "traduce el siguiente texto de inglés a español: *I like dogs*", y confiar en que el modelo de lenguaje sea capaz de realizar la traducción de manera precisa al generar palabras a partir de la instrucción.

GPT-3 [14] llevó esta noción aún más allá, al contar con aproximadamente 200 mil millones de parámetros y haber sido entrenado en un corpus gigantesco que supera los 500 mil millones de palabras. Este entrenamiento, tuvo un costo energético evaluado en torno a 4.6 millones de dólares. En esta investigación, se formaliza la propiedad emergente previamente mencionada con el término "aprendizaje en contexto" (*in-context learning*), y se proponen tres enfoques para utilizar esta propiedad.

**1. Aprendizaje sin ejemplos (*zero-shot learning*):** se le proporciona únicamente una instrucción al modelo de lenguaje en el contexto, al cual de ahora en adelante llamaremos *prompt*.

Instrucción:

"Traduce esta palabra al inglés:  
manzana →"

**2. Aprendizaje con un solo ejemplo (*one-shot learning*):** se le proporciona un solo ejemplo junto con su respuesta correcta, además de la instrucción en el *prompt*.

Instrucción:

"Traduce esta palabra al inglés:  
manzana →  
Ejemplo: zanahoria → carrot"

**3. Aprendizaje con pocos ejemplos (*few-shot learning*):** se le proporcionan unos pocos ejemplos etiquetados en el contexto de la instrucción en el *prompt*.

Además, suscitó un profundo debate en la comunidad académica acerca de si estas propiedades emergentes realmente representan un aprendizaje en base al contexto o si simplemente se deben a que el extenso corpus de entrenamiento ya contenía datos relevantes para las tareas existentes. Este cuestionamiento sigue siendo ampliamente discutido [15].

El diseño apropiado de *prompts* se ha transformado en un aspecto de vital importancia para modelos tipo GPT-3, ya que la calidad de las respuestas puede variar sustancialmente según el *prompt* empleado. En consecuencia, ha surgido una nueva disciplina conocida como "ingeniería de *prompts*", cuyo enfoque radica en descubrir métodos para redactar *prompts* de manera efectiva según la tarea que se quiere resolver.

Por otro lado, GPT-3 transformó la forma en que ingenieros e investigadores acceden a estos modelos. En la época de BERT, todos los modelos eran liberados públicamente (los pesos de la red neuronal) en plataformas abiertas como



HuggingFace<sup>2</sup>, lo que permitía su uso y adaptación por cualquier persona interesada. En contraste, GPT-3 adoptó un enfoque diferente al ser un modelo cerrado, y el acceso a sus capacidades se ofrece mediante una API de pago, lo que generó un cambio significativo en cómo se interactúa con estos modelos avanzados.

---

## Modelos de lenguaje como asistentes

---

Estos grandes modelos de lenguaje, denominados LLMs por sus siglas en inglés (Large Language Models) al ser entrenados meramente para predecir palabras (propiedad conocida como autoregresiva) no son suficientes para crear asistentes de usuario o chatbots que sean capaces de realmente asistir a los usuarios por medio del *prompt*. Pueden producir respuestas vagas, repetitivas o poco relevantes, incluso políticamente incorrectas, incluyendo discursos

de odio y prejuicios raciales o de género, para las consultas de los usuarios.

Una solución a este problema radica en llevar a cabo el ajuste fino de los modelos de lenguaje, de manera que se alineen con las intenciones del usuario. El proceso de ajuste fino implica la contratación de evaluadores humanos (también conocidos como *crowdworkers*) para interactuar con el modelo de lenguaje original y etiquetar las respuestas en función de su nivel de informatividad, seguridad, veracidad y otros criterios relevantes. Dado que esto se realiza a gran escala, posibilita la reorientación del modelo de lenguaje con el propósito de transformarlo en un asistente verdaderamente útil.

Un ejemplo de esta técnica es ChatGPT de la empresa OpenAI, que emplea un enfoque denominado aprendizaje por refuerzo a partir de retroalimentación humana (Reinforcement Learning from Human Preferences RLHF) en su proceso de ajuste fino (ver Figura 9). En este proceso, el modelo se adapta mediante una

función de preferencia que otorga una puntuación a las respuestas generadas (esta se aprende en base a etiquetadores humanos), optimizando dicha función mediante algoritmos de optimización por refuerzo (Reinforcement Learning)<sup>3</sup>. Sin embargo, es fundamental tener en cuenta que esta estrategia incrementa los costos asociados a la construcción del modelo, debido a la necesidad de contratar a un gran número de personas para que evalúen las respuestas generadas por el modelo original.

Junto a ChatGPT, diversas empresas han incursionado en una competencia por lanzar al mercado sus propios asistentes, tales como Google Bard, YouChat, entre otros. Recientemente, la empresa OpenAI presentó su producto GPT-4, cuya característica principal es su capacidad para integrar imágenes a los *prompts*. Esto le permite al modelo responder a preguntas que contienen elementos visuales, como en las pruebas estandarizadas de admisión universitaria y certificaciones en diversos campos, como

---

<sup>2</sup> <https://huggingface.co/>.

<sup>3</sup> El aprendizaje por refuerzo es una forma de aprendizaje automático en el cual un agente o programa aprende una política de acciones para maximizar una recompensa. Se usa para automatizar jugadores en juegos como el ajedrez y el Go, así como también en la robótica.



## ***El diseño apropiado de prompts se ha transformado en un aspecto de vital importancia [...], ya que la calidad de las respuestas puede variar sustancialmente según el prompt empleado.***

medicina y derecho, entre otros. No obstante, se dispone de escasa información detallada sobre los aspectos técnicos de GPT-4, ya que OpenAI únicamente proporcionó un informe técnico con limitada información sobre su construcción [16].

Sin embargo, la comunidad académica y de código abierto no ha permanecido rezagada en esta carrera. Apenas una semana después del lanzamiento de LLaMA por parte de META, sus pesos fueron filtrados al público a través de BitTorrent. Esto abrió la posibilidad para que iniciativas académicas con recursos más limitados en comparación con las grandes empresas pudieran ajustar estos modelos por su cuenta a partir de los pesos de LLaMA. Han surgido tecnologías que permiten realizar este ajuste sin requerir enormes capacidades de cómputo, mediante técnicas como Lora y QLora [17] que reducen la cantidad de bits de los pesos de la red, además de reducir la cantidad de parámetros necesarios para hacer el ajuste fino. Además, en lugar de aplicar el ajuste fino directamente sobre texto libre, se realiza sobre ejemplos de pares (instrucción, salida) que abordan diversas tareas, adaptando así el modelo en función de estas muestras.

Estos ejemplos de entrenamiento incluso se generan automáticamente a través de modelos cerrados como ChatGPT,

gracias a iniciativas como ShareGPT<sup>4</sup>, donde las personas ponen a disposición sus conversaciones con dicho modelo de forma gratuita. En esta misma línea, se han liberado modelos como Alpaca<sup>5</sup> y Vicuna<sup>6</sup>. En julio de 2023, META lanzó su segunda generación de LLMs abiertos bajo el nombre de LLaMA 2, con distintas versiones de 7, 13 y 70 mil millones de parámetros respectivamente [18].

Además, dado la enorme cantidad de LLMs que salen todas las semanas, la comunidad académica ha ideado métodos ingeniosos para evaluarlos, como la “Chatbot Arena”<sup>7</sup>, una plataforma donde diversos modelos de lenguaje compiten para resolver tareas y reciben puntuaciones de manera análoga al sistema ELO utilizado en el ajedrez.

---

## **Conclusiones**

---

Acabamos de recorrer la historia de los modelos de lenguaje, desde los simples modelos de  $n$ -gramas hasta los complejos LLMs basados en Transformers capaces de resumir, traducir, generar programas e inventar historias.

Tanto la comunidad científica como la sociedad civil han señalado diversos riesgos vinculados a esta tecnología [19]. Uno de estos riesgos notables es

la alucinación, que se refiere a la capacidad de estos modelos para generar información falsa. Además, emerge con fuerza la preocupación por la equidad, dado que estos modelos pueden agravar los sesgos presentes en los datos de entrenamiento, lo que lleva a la propagación de prejuicios relacionados con género, raza, orientación sexual y otras minorías.

La violación de derechos de autor emerge como un riesgo legal importante, ya que los LLMs pueden vulnerar las leyes de propiedad intelectual al reproducir contenido sin la debida autorización. Además, los altos costos asociados con el entrenamiento de los LLMs plantean inquietudes acerca de la posibilidad de una concentración monopólica. La transparencia también se ve comprometida, dado que la complejidad intrínseca de estos modelos dificulta la explicación de sus respuestas. Por último, el proceso de entrenamiento de estos modelos conlleva un alto impacto ambiental con una contribución significativa a las emisiones de carbono.

Dado lo acelerado que ha sido el progreso de los modelos de lenguaje en el último tiempo, es muy difícil predecir qué estarán haciendo en el futuro. Sí podemos decir con certeza que cada vez abundarán más modelos generativos de texto y otros tipos de datos (imágenes, video, código), como también programas (o agentes)<sup>8</sup> que hacen uso automatizado de estos modelos para tomar acciones como comprar pasajes, invertir, agendar consultas médicas, entre otras posibilidades que aún no nos imaginamos. ■

---

4 <https://sharegpt.com/>.

5 <https://crfm.stanford.edu/2023/03/13/alpaca.html>.


6 <https://lmsys.org/blog/2023-03-30-vicuna/>.

7 <https://lmsys.org/blog/2023-05-03-arena/>.

8 En esta línea destacamos Langchain, un biblioteca de programación para generar aplicaciones basadas en LLMs: <https://python.langchain.com>.



## REFERENCIAS

- [3] Shannon, C. E. (1951). Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64.
- [4] Chomsky, N. (2009). Syntactic structures. In *Syntactic Structures*. De Gruyter Mouton.
- [5] Rosenblatt, F. (1957). *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory.
- [6] Ivakhnenko, A., & Lapa, V. G. (1965). Cybernetic Predicting Devices. CCM Information Corporation. *First working Deep Learners with many layers, learning internal representations*.
- [7] Bengio, Y., Ducharme, R., and Vincent, P. (2000). A neural probabilistic language model. *Advances in neural information processing systems*, 13.
- [8] Harris, Z. (1954). Distributional structure. *Word*, 10(23): 146–162.
- [9] Firth, J.R. (1957). A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*, pp. 1-32. Oxford: Philological Society. Reprinted in F.R. Palmer (ed.), *Selected Papers of J.R. Firth 1952-1959*, London: Longman (1968).
- [10] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.
- [11] Mikolov, T., Karafiát, M., Burget, L., Cernocký, J., & Khudanpur, S. (2010, September). Recurrent neural network based language model. In *Interspeech*, Volumen 2, No. 3, pages 1045–1048.
- [12] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- [13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gómez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.
- [14] Kenton, J. D. M.-W. C. and Toutanova, L. K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.
- [15] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9.
- [16] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- [17] Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., ... & Fedus, W. (2022). Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.
- [18] OpenAI (2023). Gpt-4 technical report.
- [19] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). Qlora: Efficient finetuning of quantized LLMs. *arXiv preprint arXiv:2305.14314*.
- [20] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- [21] Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021, March). On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency* (pp. 610–623).