

Un recorrido por los Modelos de Lenguaje Desde Shannon a GPT-4

Felipe Bravo Márquez



dcc

CIENCIAS DE LA COMPUTACIÓN
UNIVERSIDAD DE CHILE

CEN

CENTRO NACIONAL DE INTELIGENCIA ARTIFICIAL



Millennium Institute
Foundational
Research on Data

RELELA

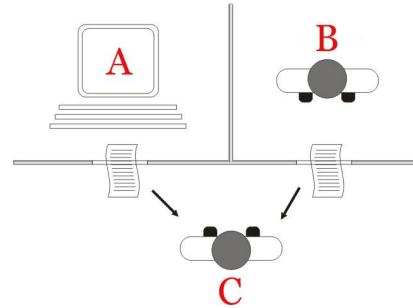
Representations for
Learning and Language

Contexto Histórico

El Diálogo y la Inteligencia Artificial

1950 - Turing Test: ¿Se puede crear una máquina que sea indistinguible de una persona en una conversación?

1964 - Eliza por Joseph Weizenbaum:
agente de conversación que simula un psicoterapeuta en base a reglas.



ELIZA—A Computer Program
For the Study of Natural Language
Communication Between Man
And Machine

JOSEPH WEIZENBAUM
Massachusetts Institute of Technology, Cambridge, Mass.*



Shannon y Chomsky

1950 - Claude Shannon realiza los primeros estudios de modelar el lenguaje escrito de manera estadística y predictiva.

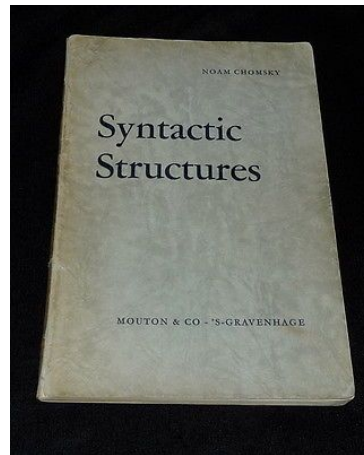
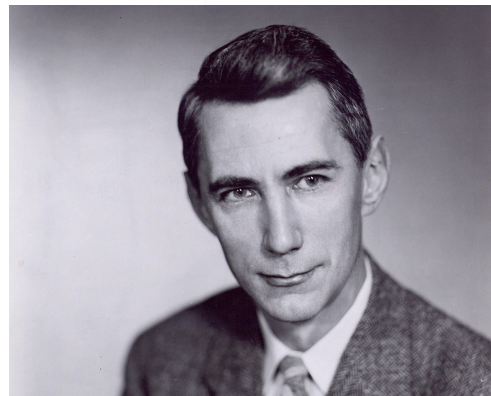
1957 - Noam Chomsky cuestiona la capacidad de modelos estadísticos para identificar la gramática del lenguaje.

‘The notion “grammatical in English” cannot be identified in any way with the notion “high order of statistical approximation to English” ‘.

Prediction and Entropy of Printed English

By C. E. SHANNON

(Manuscript Received Sept. 15, 1950)



Primeros Modelos de Lenguaje

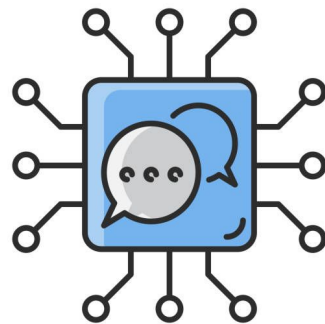
Modelo de Lenguaje

Modelo que asigna una probabilidad a cualquier oración:

- $P(\text{El perro ladra}) = 0.01$
- $P(\text{Me gusta el fútbol}) = 0.001$
- $P(\text{gato verde duerme fuerte}) = 0.000001$

Idea: el modelo le asigna probabilidad mayor a oraciones **fluidas** (que hacen sentido, que son gramaticalmente correctas)

Idea: estimar esta función de probabilidad a partir de texto (corpus)



Modelo de Lenguaje

El modelo de lenguaje ayuda a modelos que **genera** textos a distinguir entre oraciones buenas y malas:

- Traducción automática
- Reconocimiento del habla
- Generación automática de resúmenes
- Chatbots

P(recognize speech) vs P(wreck a nice beach)



Modelo de Lenguaje de n-gramas

- Puedo modelar una oración como una secuencia de palabras

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

- Puedo factorizar la probabilidad de una secuencia de palabras como un producto de probabilidades condicionales

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

Modelo de Lenguaje de n-gramas

- Para facilitar la estimación, limitamos la memoria del contexto

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

- Luego la probabilidad de la oración queda como:

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

Modelo de Lenguaje de n-gramas

- La probabilidad de generar una palabra dado la anterior se estima así:

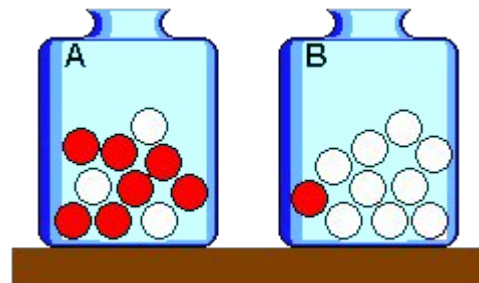
$$p(w_i|w_{i-1}) = \frac{\text{count}(w_{i-1}, w_i)}{\text{count}(w_{i-1})}$$

- Ejemplo:

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

Modelos de Lenguaje son Generativos

- Un LM puede generar oraciones haciendo un muestreo secuencial sobre las probabilidades.
- Esto es análogo a sacar bolitas (palabras) de una urna donde éstas tienen un tamaño proporcional a sus frecuencias relativas.
- También podría sacar siempre la palabra más probable. Lo que equivale a **predecir la palabra siguiente**.

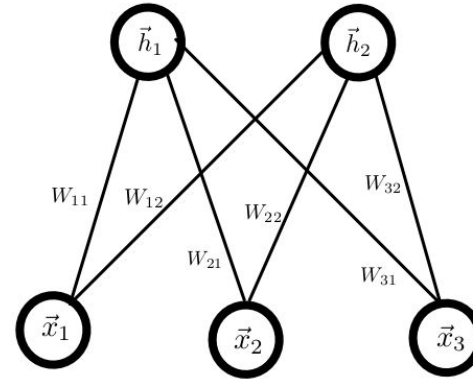
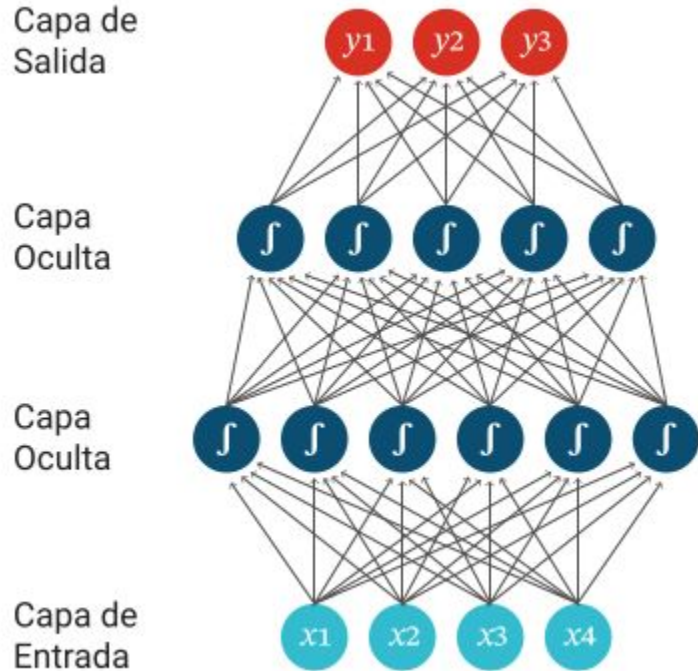


Limitaciones de Modelos de Lenguaje de N-gramas

- No capturan dependencias largas ni aprovechan contextos similares (sinónimos).
- Por ejemplo, consideremos los contextos:
 - c1) Después de comer cereales
 - c2) Luego de desayunar avena
- Esperaríamos que las distribuciones de probabilidad $p(w|c1)$ y $p(w|c2)$ fueran similares
- Pero c1 y c2 casi no comparten palabras
- Como los modelos de n-gramas que se limitan a contar frecuencia de palabras no pueden capturar estas similitudes entre contextos.

Modelos de Lenguaje Neuronales

Redes Neuronales (Rosenblatt 1957, Ivakhnenko and Lapa 1965)



$$\vec{x} = [\vec{x}_1, \vec{x}_2, \vec{x}_3] \quad W = \begin{pmatrix} W_{1,1} & W_{1,2} \\ W_{2,1} & W_{2,2} \\ W_{3,1} & W_{3,2} \end{pmatrix}$$

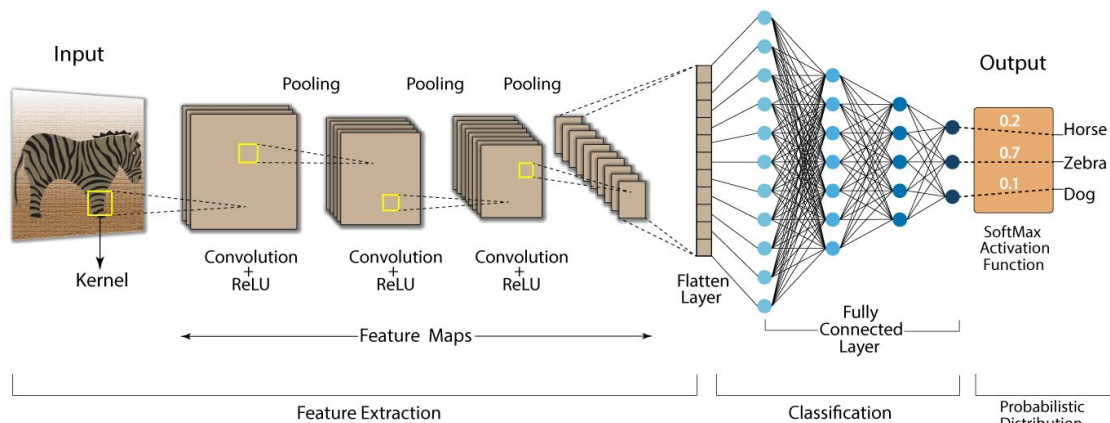
$$\vec{h} = \vec{x}W$$

$$\vec{x}W = [\vec{x}_1 * W_{11} + \vec{x}_2 * W_{21} + \vec{x}_3 * W_{31}, \vec{x}_1 * W_{12} + \vec{x}_2 * W_{22} + \vec{x}_3 * W_{32}]$$

$$\vec{h} = [\vec{h}_1, \vec{h}_2]$$

Deep Learning (años 2006)

Convolution Neural Network (CNN)



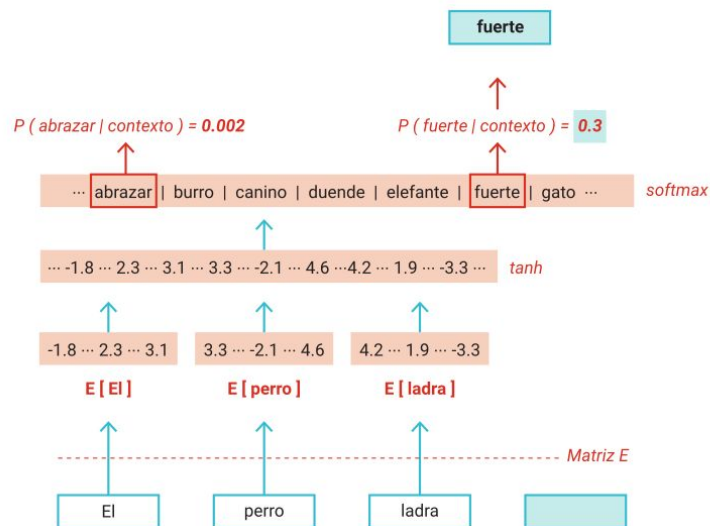
BIG DATA



Modelo de Lenguaje Neuronal (año 2000)

A Neural Probabilistic Language Model

Yoshua Bengio*, Réjean Ducharme and Pascal Vincent
Département d'Informatique et Recherche Opérationnelle
Centre de Recherche Mathématiques
Université de Montréal
Montréal, Québec, Canada, H3C 3J7
yoy,ducharme,vincentp}@iro.umontreal.ca



Matriz de embeddings

$|V| \times d$

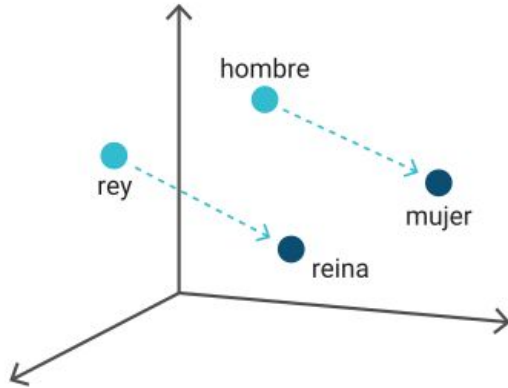
$E =$	-6.6	4.3	-2.4	← abrazar
	3.6	-1.9	4.1	← canino
	-1.8	2.3	3.1	← el
	2.5	3.4	-2.7	← fuerte
	5.0	-3.4	1.4	← grueso
	4.2	1.9	4.2	← ladra
	3.3	-2.1	4.6	← perro
	-2.6	1.5	1.6	← zancudo

- Palabras parecidas reciben vectores latentes o “**embeddings**” similares.
- Hipótesis Distribucional.
- Aún es caro modelar contextos largos.

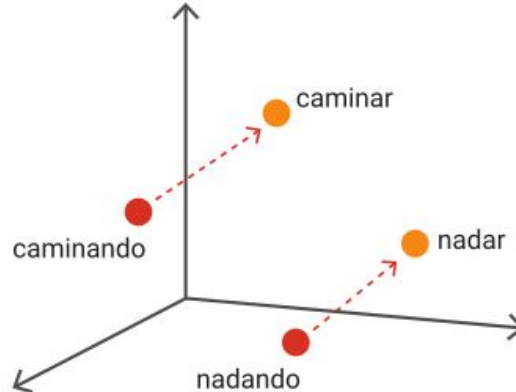
Word2Vec (año 2013)

- Tuvieron que pasar 13 años para que los LMs neuronales se usaran de forma masiva con el lanzamiento de Word2Vec (cientos de parámetros).

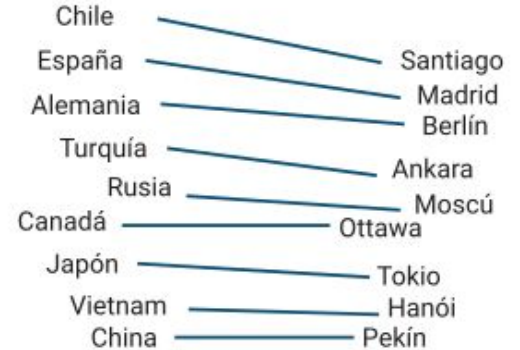
Masculino / Femenino



Forma verbal



País / Capital



Language Models con Redes Recurrentes 2010

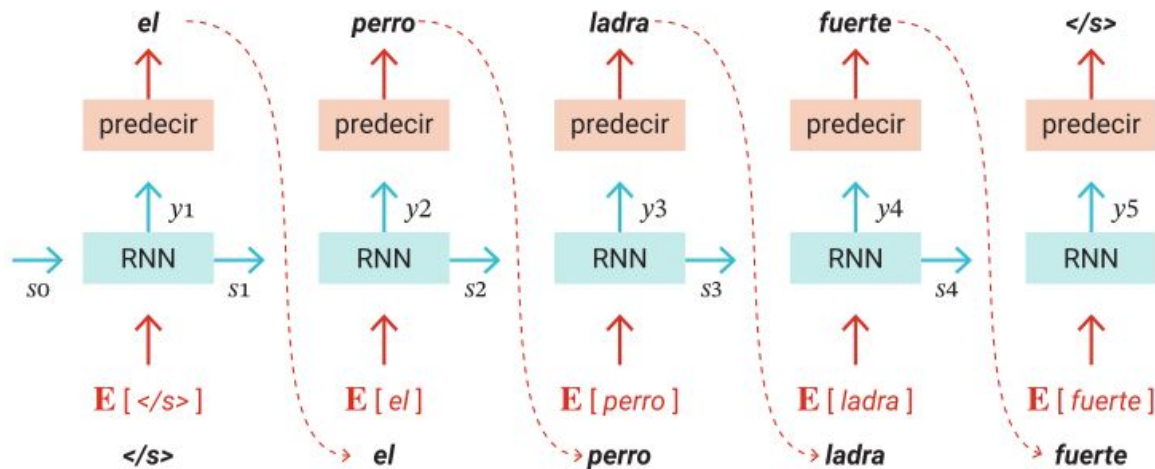
Recurrent neural network based language model

Tomáš Mikolov^{1,2}, Martin Karafiát¹, Lukáš Burget¹, Jan “Honza” Černocký¹, Sanjeev Khudanpur²

¹Speech@FIT, Brno University of Technology, Czech Republic

² Department of Electrical and Computer Engineering, Johns Hopkins University, USA

{imikolov,karafiata,burget,cernocky}@fit.vutbr.cz, khudanpur@jhu.edu



- Usar redes recurrentes permite al LM incorporar contextos más largos.
- “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 “
- Problema: caro de paralelizar.

Language Models con Redes Recurrentes producen texto de mejor calidad (2016)

Marks live up in the club comes the handed up moved to a brief d
The man allowed that about health captain played that alleged to
If you have for the past said the police say they goting ight n
However , he 's have constance has been apparents are about home
The deal share is dipled that a comments in Nox said in one of t
Like a sport released not doing the opposition overal price tabl

Fuente:

<https://arxiv.org/pdf/1706.01399.pdf>

Modelos de Lenguaje son multi-tarea

Etiquetado de Datos en NLP

- En NLP, tareas específicas como traducción automática, clasificación de documentos y generación de resúmenes se basan en **datos etiquetados**.
- El etiquetado de datos es un **proceso costoso** que implica la intervención de etiquetadores humanos.
- Los conjuntos de entrenamiento para estas tareas suelen ser pequeños, con solo miles de ejemplos.
- Esto afecta la capacidad de los modelos para **generalizar** a datos no vistos en el entrenamiento.
- Por otro lado, Los modelos de lenguaje neuronales **no** requieren texto etiquetado para el entrenamiento.

¿Qué aprendemos al predecir palabras?

- Tarea de predecir palabras: Requiere que el modelo de lenguaje aprenda **sintaxis, semántica y conocimiento general**.
- Ejemplos de contextos:
 - "Abróchate el cordón de tus...": El modelo debe entender que "zapatos" o "zapatillas" llevan **cordones**.
 - "Este regalo es para mi...": La red neuronal debe reconocer que "novio", "amigo" o "mamá" son sustantivos adecuados, implicando **conocimiento gramatical**.
 - "Chile queda en...": Para predecir "Sudamérica", el modelo necesita **conocimiento general** de geografía.



ELMo (2018): Embeddings from Language Models

- Language Models basados en redes recurrentes con 100 millones de parámetros producen **embeddings contextualizados**.

- Estos manejan mejor la **polisemia** y se pueden usar para modelos en **otras tareas**.

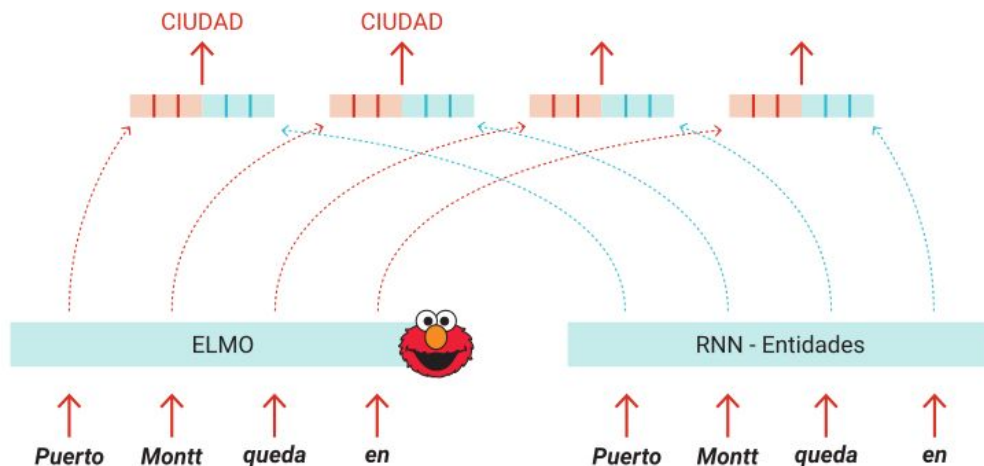
Deep contextualized word representations

Matthew E. Peters[†], Mark Neumann[†], Mohit Iyyer[†], Matt Gardner[†],
{matthewp, markn, mohiti, mattg}@allenai.org

Christopher Clark^{*}, Kenton Lee^{*}, Luke Zettlemoyer^{†*}
{csquared, kentonl, lsz}@cs.washington.edu

[†]Allen Institute for Artificial Intelligence

^{*}Paul G. Allen School of Computer Science & Engineering, University of Washington



Transformer (2017)

2017 - Vaswani et. al. proponen el **Transformer**, un tipo de red neuronal basada en mecanismos de (auto)atención.

A diferencia de las RNNs que procesan inputs de forma secuencial, los Transformers son paralelizables.

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

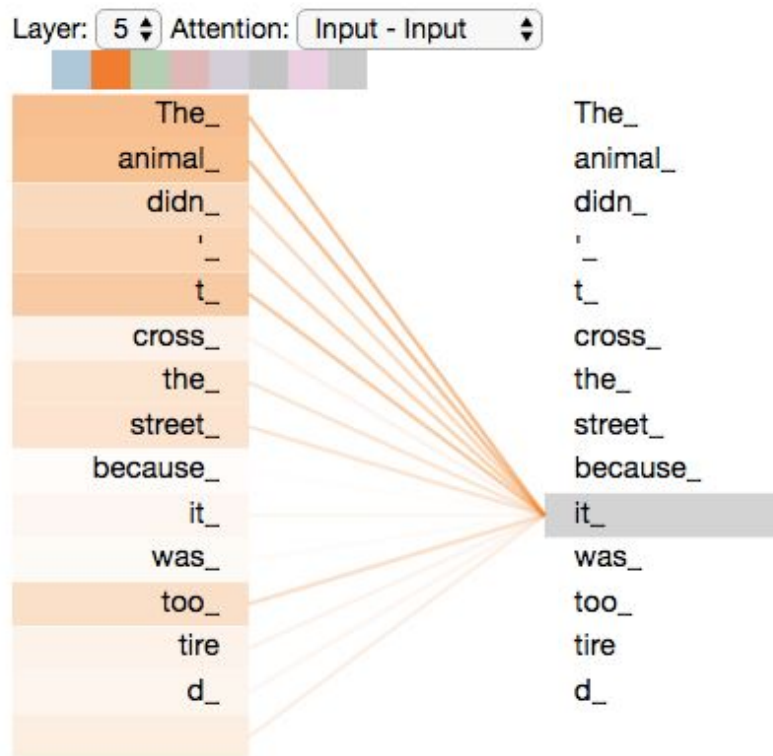
Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

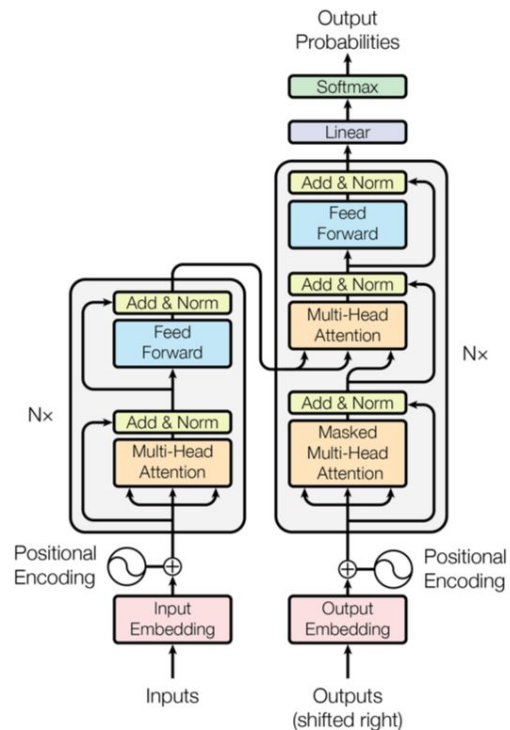
Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaizer@google.com

Illia Polosukhin* †
illia.polosukhin@gmail.com



Transformer (2017)



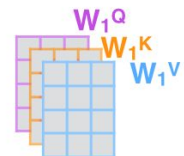
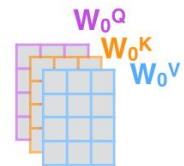
1) This is our input sentence*

Thinking
Machines

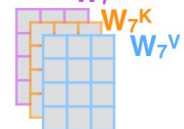
2) We embed each word*



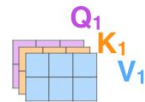
3) Split into 8 heads. We multiply X or R with weight matrices



...



4) Calculate attention using the resulting $Q/K/V$ matrices



...



5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer



...



* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

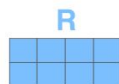


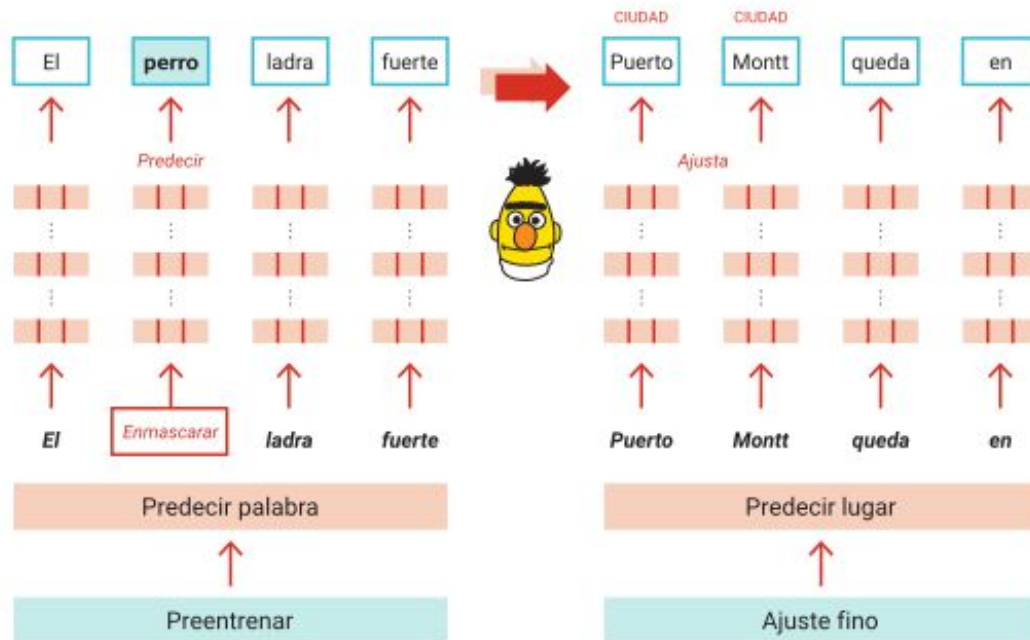
Figure 1: The Transformer - model architecture.

BERT (2019)

- 2019- BERT (replica idea de ELMO pero usando un Transformer en vez de una RNN) con 335 millones de parámetros
- La auto-atención de BERT se puede usar para reemplazar la recursión de la RNN.
- Se obtienen embeddings contextualizados de forma mucho más **escalable** (paralelismo) en cantidad de parámetros y datos.



Fine-Tuning



Un Language Model pre-entrenado puede **fine-tunearse** para resolver cualquier tarea de NLP con ganancias importantes.

Texto Generado por LM basado en Transformers son muy superiores a lo anterior

SYSTEM PROMPT (HUMAN-WRITTEN) *In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*

MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES) The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. ...

Ejemplo generado por GPT-2

Los Modelos de Lenguaje son Multi-task Learners

- Hasta ahora para adaptar un LM a otra tarea (ej: clasificación, traducción, respuesta a preguntas) necesito **adaptar** la **arquitectura** para la tarea objetivo.
- Radford et. al proponen simplemente codificar la tarea **(instrucción)** y el ejemplo en el contexto del modelo de lenguaje.
- Por ejemplo: $P(X| \text{“traducir esto al inglés: me gusta la pizza”})$

Language Models are Unsupervised Multitask Learners

Alec Radford ^{* 1} Jeffrey Wu ^{* 1} Rewon Child ¹ David Luan ¹ Dario Amodei ^{** 1} Ilya Sutskever ^{** 1}

Cambio de Fase: GPT-3 (2020)

- GPT-3: otro LM basado en el Transformer con casi 200 mil millones de parámetros y entrenado sobre un corpus gigante (casi 500 mil millones de palabras).
- **In-context learning**: pueden resolver otras tareas de NLP usando **zero-shot y few-shot learning**.
- Toda la magia está en el **prompt**.
- No es necesario hacer fine-tuning para resolver otros problemas.



Nuevos Paradigmas de Aprendizaje

Aprendizaje sin ejemplos
(zero-shot learning)

Instrucción:

```
"Traduce esta palabra al  
inglés: manzana →"
```

Aprendizaje con un ejemplo
(one-shot learning)

Instrucción:

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

Aprendizaje con unos pocos
ejemplos (few-shot
learning)

Instrucción:

```
"Traduce esta palabra al inglés:  
manzana →  
Ejemplos:  
- zanahoria → carrot  
- limón → lemon  
- lechuga → lettuce  
- pera → pear"
```

Esto es un cambio de paradigma **muy profundo**.

Prompt Engineering

Rule #1 – Instructions at beginning and ### or "" to separate instructions or context

✗ Rewrite the text below in more engaging language.
{your input here}

✓ Rewrite the text below in more engaging language.
Text: ""
{your input here}
""

Rule #2 – Be specific and detailed about the desired context, outcome, length, format, and style.

✗ Write a short story for kids

✓ Write a funny soccer story for kids that teaches the kid that persistence is key for success in the style of Rowling.

Rule #3 – Give examples of desired output format

✗ Extract house pricing data from the following text.
Text: ""
{your text containing pricing data}
""

✓ Extract house pricing data from the following text.
Desired format: ""
House 1 | \$1,000,000 | 100 sqm
House 2 | \$500,000 | 90 sqm
... (and so on)
""
Text: ""
{your text containing pricing data}
""

Rule #4 – First try without examples, then try giving some examples.

✓ Extract brand names from the text below.

Text: {your text here}

Brand names:

✓ Extract brand names from the texts below.

Text 1: Finxter and YouTube are tech companies. Google is too.
Brand names 2: Finxter, YouTube, Google
###

Text 2: If you like tech, you'll love Finxter!
Brand names 2: Finxter
###

Text 3: {your text here}
Brand names 3:

Rule #5 – Fine-tune if Rule #4 doesn't work

Fine-tuning improves model performance by training on more examples, resulting in higher quality results, token savings, and lower latency requests.

ChatGPT can intuitively generate plausible completions from few examples, known as **few-shot learning**.

Fine-tuning achieves better results on various tasks without requiring examples in the prompt, saving costs and enabling lower-latency requests.

Example Training Data

```
{"prompt": "<input>", "completion": "<ideal output>"}  
{"prompt": "<input>", "completion": "<ideal output>"}  
{"prompt": "<input>", "completion": "<ideal output>"}  
...
```

Rule #6 – Be specific. Omit needless words.

✗ ChatGPT, write a sales page for my company selling sand in the desert, please write only a few sentences, nothing long and complex

✓ Write a 5-sentence sales page, sell sand in the desert.

Rule #7 – Use leading words to nudge the model toward a pattern

✗ Write a Python function that plots my net worth over 10 years for different inputs on the initial investment and a given ROI

✓ # Python function that plots net worth over 10 years for different inputs on the initial investment and a given ROI

```
import matplotlib
```

```
def plot_net_worth(initial, roi):
```

Chatgpt Prompting Cheatsheet - OpenAI

Chain-of-thought Prompting

- Chain-of-thought prompting es un mecanismo para ayudar al modelo de lenguaje a razonar su respuesta
- Idea: Aumentar cada ejemplo few-shot con una cadena de pensamiento para encontrar la respuesta correcta [Wei et al., 2022].

Chain-of-thought prompting

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✅

Modelos de Lenguaje como Asistentes

Modelos de lenguaje como asistentes de usuario (o chatbots)

- Los Modelos Autorregresivos de Lenguaje Grande **no están alineados** con la intención del usuario [Ouyang et al., 2022].
- Solución: alinear el modelo de lenguaje con la intención del usuario mediante **fine-tuning**.

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION GPT-3

Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.

LaMDA: Language Models for Dialog Applications (2022)

LaMDA es un modelo de lenguaje basado en Transformer optimizado para el diálogo de **dominio abierto**.

Es pre-entrenado inicialmente de la misma forma que los modelos de lenguaje tradicionales (predecir palabras) con un fuerte foco en datos de diálogo.

Luego es ajustado (**fine-tuned**) para generar respuestas respecto a varios otros criterios.

#parámetros	#palabras con que se entrenó
137 mil millones	1.56 billones



Crowd-sourcing: La Clave del Éxito

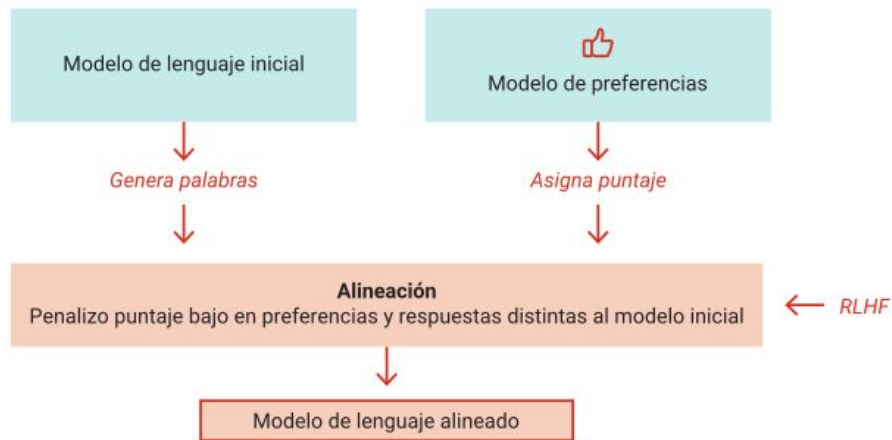
Para poder ajustar LaMBDA a todos esos criterios se trabajó con un alto número de **crowd-workers**.

Estas son personas que **etiquetaron manualmente** conversaciones del modelo pre-entrenado.



ChatGPT (2022)

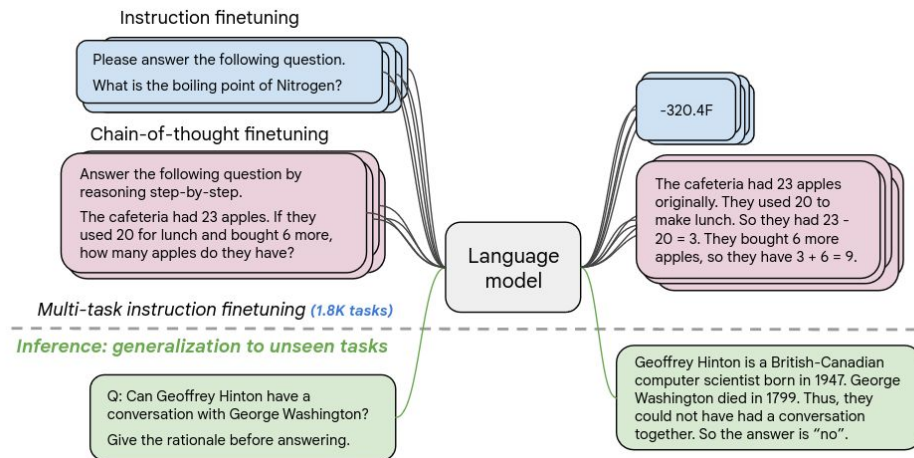
- Modelo similar a Lambda lanzado por **OpenAI** a fines del 2022.
- También usa Crowdsourcing para mejorar sus respuestas pero su proceso de fine-tuning usa **Reinforcement Learning** (paradigma distinto).
- En particular usa **Reinforcement Learning from Human Feedback (RLHF)**.
- **Idea:** Construir un modelo de preferencias (que le asigna puntaje a una oración generada) y ajustar el LM para considerarlas.



Fine-Tuning de Instrucciones

- Una forma más eficiente de afinar los modelos de lenguaje de gran tamaño es hacer **Fine-Tuning de Instrucción** [Chung et al., 2022].

- Idea: recopilar ejemplos de pares (instrucción, salida) a través de muchas tareas y ajustar el modelo de lenguaje con esos datos.



- Evaluar en otras tareas.
- Esta idea está impulsando fuertemente el desarrollo de modelos de lenguaje open source (Alpaca, Vicuna, Falcon, Mistral, Llama).

GPT-4, Gemini (2023)

- Últimos LM de OpenAI y Google
- Capaces de incluir **imágenes** en el prompt.
- Sigue siendo un Transformer LM.
- Capaz de aprobar exámenes en diversas disciplinas pudiendo procesar las imágenes de las preguntas.

Example of GPT-4 visual input:

User What is funny about this image? Describe it panel by panel.



Source: <https://www.reddit.com/r/hmmm/comments/ubab5v/hmmm/>

GPT-4 The image shows a package for a "Lightning Cable" adapter with three panels.

Panel 1: A smartphone with a VGA connector (a large, blue, 15-pin connector typically used for computer monitors) plugged into its charging port.

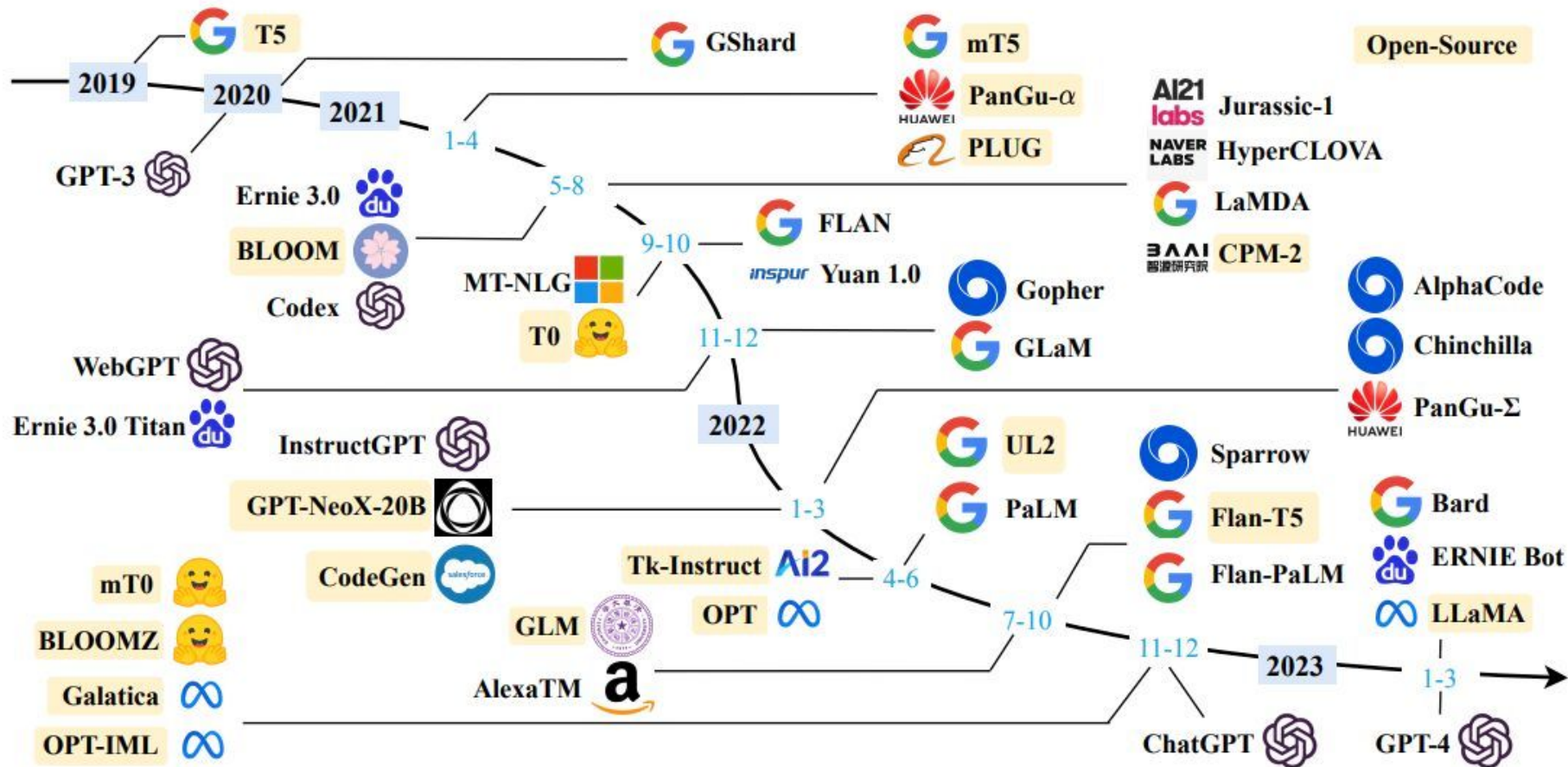
Panel 2: The package for the "Lightning Cable" adapter with a picture of a VGA connector on it.

Panel 3: A close-up of the VGA connector with a small Lightning connector (used for charging iPhones and other Apple devices) at the end.

The humor in this image comes from the absurdity of plugging a large, outdated VGA connector into a small, modern smartphone charging port.

Table 3. Example prompt demonstrating GPT-4's visual input capability. The prompt consists of a question about an image with multiple panels which GPT-4 is able to answer.

Fuente: <https://arxiv.org/pdf/2303.08774.pdf>



Fuente: A Survey of Large Language Models (<https://arxiv.org/abs/2303.18223>)

Conclusiones

- El crecimiento en volumen y capacidades de los modelos de lenguaje ha tenido una **aceleración impresionante**.
- Es muy difícil predecir qué estarán haciendo en el futuro.
- ¿Qué podemos predecir con certeza?
 - Abundarán modelos generativos para **múltiples formatos** (texto, código, imagen, video, realidades virtuales).
 - Abundarán **agentes/programas** que actúen y tomen decisiones mediante la interacción con estos modelos (citas médicas, inversiones, viajes).