



Explorando los Transformers y las Limitaciones en los Modelos de Lenguaje

Danae Sanchez, Post-doc, DIKU, Junio 2024

Hi!



Los modelos de lenguaje de gran tamaño (LLMs) han demostrado capacidades excepcionales

Escribe una lista de tres actividades en la Ciudad de México.



La Ciudad de México es una ciudad llena de vida y cultura. Aquí te dejo tres actividades que puedes disfrutar en la capital mexicana:

1. ****Visitar el Zócalo y la Catedral Metropolitana****: El Zócalo es el corazón de la ciudad y es un lugar donde se encuentran importantes monumentos y edificios históricos, como la Catedral Metropolitana, el Palacio de Bellas Artes y el Ayuntamiento. Puedes disfrutar de la arquitectura colonial, la música y la comida tradicional en este lugar emblemático.

2. ****Explore el Museo Nacional de Antropología****: El Museo Nacional de Antropología es uno de los museos más importantes de la ciudad y es dedicado a la cultura y la historia de México. La colección del museo incluye artefactos y objetos de la cultura maya, azteca y olmeca, entre otros. Puedes ver exhibiciones sobre la vida y la cultura de los antiguos mexicanos y aprender sobre la historia de la ciudad.

3. ****Visitar el Mercado de San Juan****: El Mercado de San Juan es un lugar emocionante donde puedes encontrar una variedad de productos y artesanías tradicionales mexicanas. Puedes probar comida local, comprar ropa y souvenirs, y disfrutar de la energía y el ambiente vibrante del mercado. Es un lugar ideal para encontrar regalos únicos y auténticos para llevar a tus seres queridos.

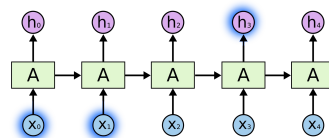
Espero que disfrutes estas actividades en la Ciudad de México. ¡Disfruta tu viaje!

Avances en Procesamiento de Lenguaje Natural (NLP)

← 2017:



- BoW, TF.IDF, Word2Vec, RNNs, LSTMs, Seq2seq



2017: The Transformer



2018: GPT, BERT



2019: GPT-2

2020: GPT-3

2022: ChatGPT



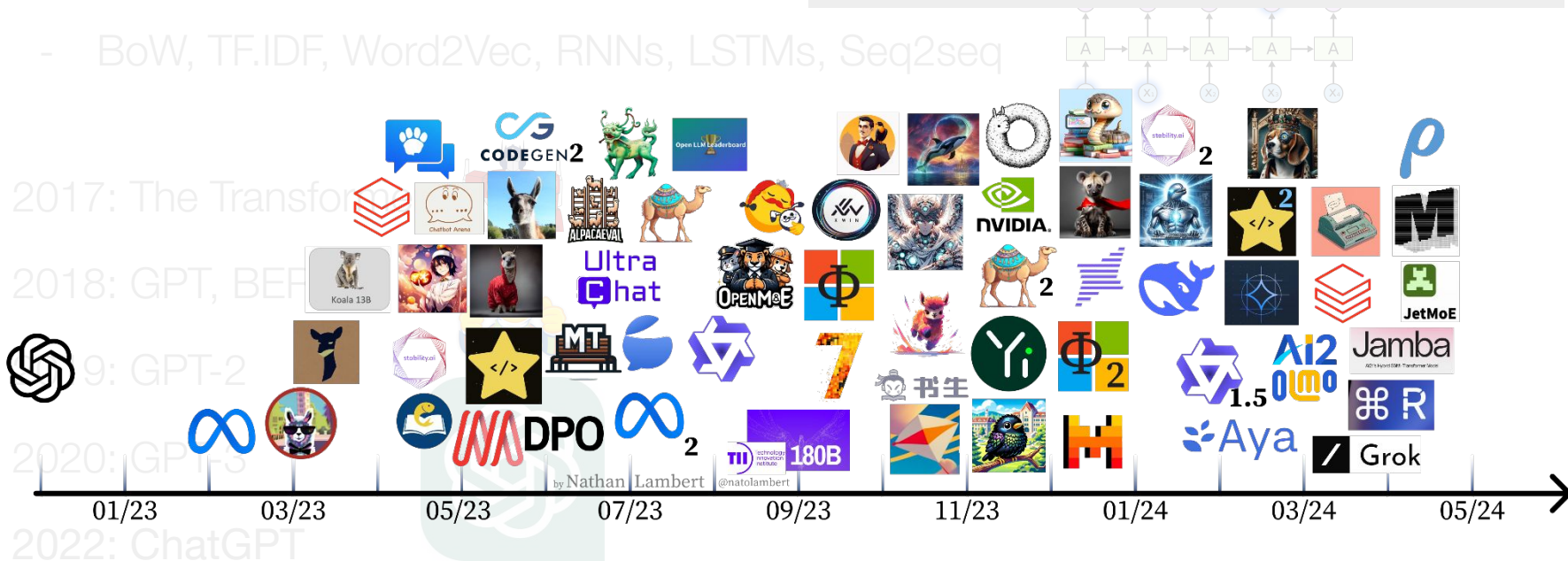
Avances en NLP

← 2017:

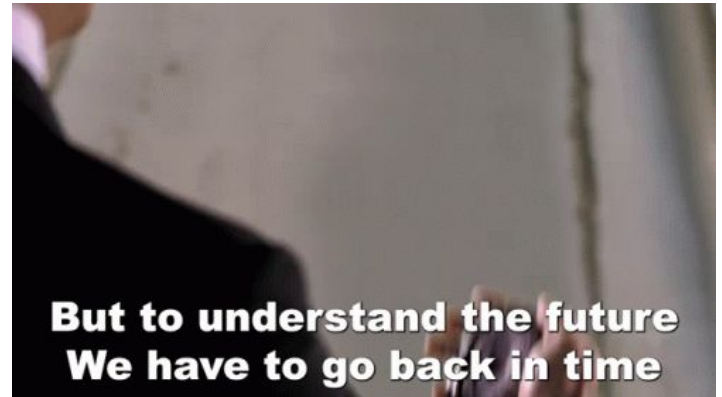


Modelos robustos entrenados con
instruction tuning y/o RLHF

- BoW, TF.IDF, Word2Vec, RNNs, LSTMs, Seq2seq



¿Cómo llegamos aquí?



Esquema

1. Modelos Neuronales de Lenguaje
2. Limitaciones de los Modelos de Lenguaje de gran tamaño
3. Reflexiones Finales

1. Modelos Neuronales de Lenguaje

Modelado de Lenguaje

Distribuciones de probabilidad sobre oraciones (es decir, secuencias de palabras)

$$P(W) = P(w_1, w_2, \dots, w_n)$$

El modelado de lenguaje implica crear un modelo probabilístico para predecir la próxima palabra en una secuencia dadas las palabras precedentes.

$$P(w_n | w_1, w_2, \dots, w_{n-1})$$

$$P(\text{"Hoy es sabado"}) > P(\text{"Hoy sabado es"}) \quad \bigg| \quad P(\text{"Hoy es sabado"}) > P(\text{"Hoy es tacos"})$$

N-gram Language Models

Regla de la Cadena de Probabilidad

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

N-gram Models (N=3)

$$P(w_n|w_1, w_2, \dots, w_{n-1}) \approx P(w_n|w_{n-2}, w_{n-1})$$

	Word	↓ Count
1	one of the	303,265
2	as well as	291,495
3	part of the	151,966
4	I do n't	148,894
5	a lot of	145,674

N-gram Language Models

Modelos N-gram (N=3)

$$P(w_n | w_1, w_2, \dots, w_{n-1}) \approx P(w_n | w_{n-2}, w_{n-1})$$

Estimando Probabilidades

$$P(w_n | w_{n-2}, w_{n-1}) = \frac{\text{Count}(w_{n-2}, w_{n-1}, w_n)}{\text{Count}(w_{n-2}, w_{n-1})}$$

	Word	↓ Count
1	one of the	303,265
2	as well as	291,495
3	part of the	151,966
4	I do n't	148,894
5	a lot of	145,674

Limitaciones de los Modelos de Lenguaje N-gram

QUEREMOS DEPENDENCIAS MÁS LARGAS

```
sentences = [  
    "New York has amazing food",  
    "New York has amazing  
    nightlife",  
    "Paris has amazing art",  
    "Paris has amazing culture"]
```

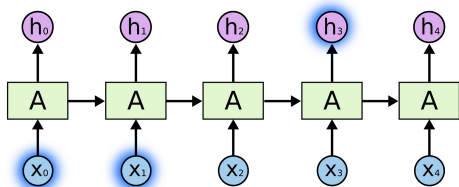
```
# Predict the next word  
context = ("has", "amazing")  
prediction = model.predict(*context)  
print(f"Predicted next word after  
'{context[0]} {context[1]}': {prediction}")
```

El modelo selecciona al azar: "comida", "vida nocturna", "arte" o "cultura" ya que cada palabra tiene una probabilidad igual (1/4 o 25%) de ser elegida porque todas aparecen una vez en los datos de entrenamiento.

Modelos Neuronales de Lenguaje

Los modelos neuronales de lenguaje utilizan redes neuronales para captar las dependencias entre las palabras.

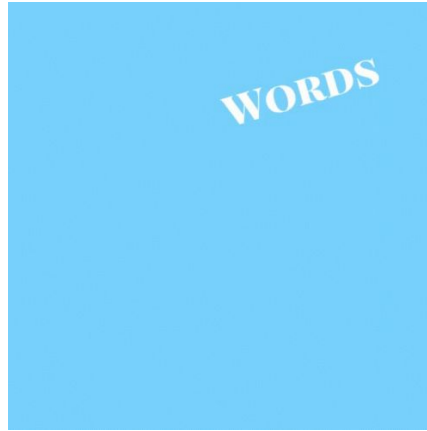
Recurrent Neural Networks (RNNs)



Transformers



¿Cómo podemos representar las palabras?



Representaciones de Bolsa de Palabras (BoW)

- **Representaciones Term Frequency (TF)**
- *Cada palabra es una entrada única*
- ***El orden de las palabras no se captura.***

[2, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1]

#1 #5 #4 #10 #15 #3 #1 #17 #20

The quick brown fox jumps over the lazy dog

Limitaciones de las Representaciones BoW

El orden de las palabras no se captura

No podemos distinguir entre las mismas palabras dispuestas de manera diferente.

1. "El perro se comió la tarea del niño."
2. "El niño se comió la tarea del perro."

```
Vocabulario: ['comió', 'del', 'el', 'la', 'niño', 'perro', 'se', 'tarea']  
Representación BoW de la oración 1: [1 1 1 1 1 1 1 1]  
Representación BoW de la oración 2: [1 1 1 1 1 1 1 1]
```

Embeddings de palabras: representando palabras por su contexto

Tejuino



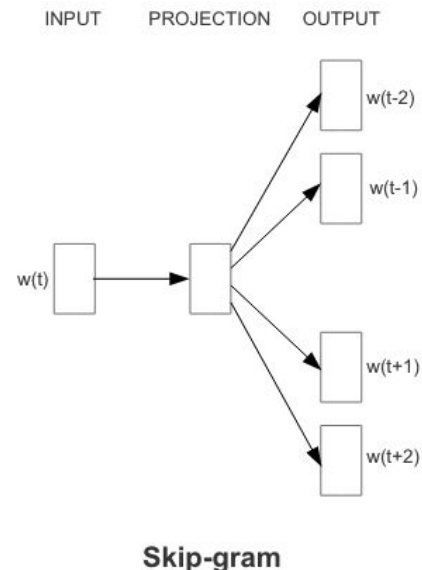
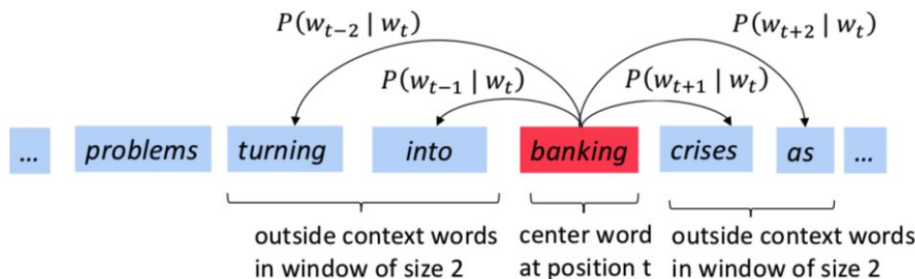
Es una bebida refrescante a base de maíz y de dulce de caña de azúcar (piloncillo).

1. Una botella de ___ está sobre la mesa.
2. No tengas ___ antes de conducir.
3. Hacemos ___ con maíz.

Embeddings de palabras: representando palabras por su contexto

Modelo Skip-Gram:

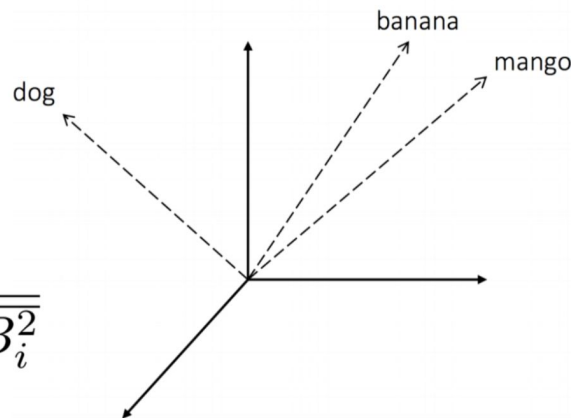
- Predice las palabras de contexto dada una palabra objetivo.
- Pre-entrena la representación de palabras (vectores) en corpus de gran tamaño.



Word2Vec

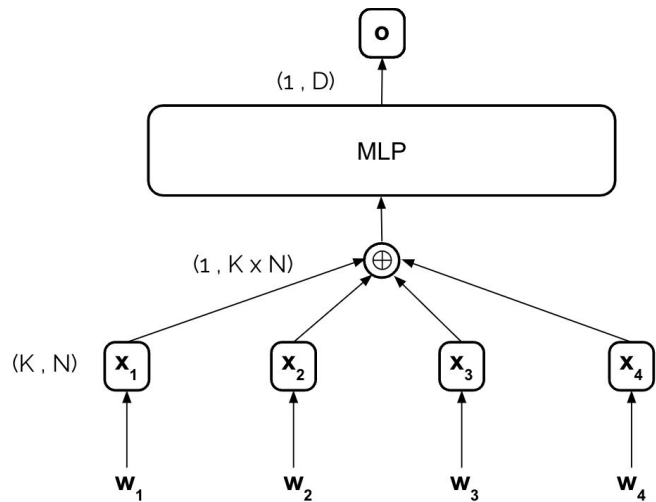
- Un **espacio** de que refleja la **semántica** de las palabras.
- **Similitud**: "Banana" está más cerca de "mango" que de "perro".

$$\begin{aligned}\cos(\theta) &= \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \\ &= \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}\end{aligned}$$



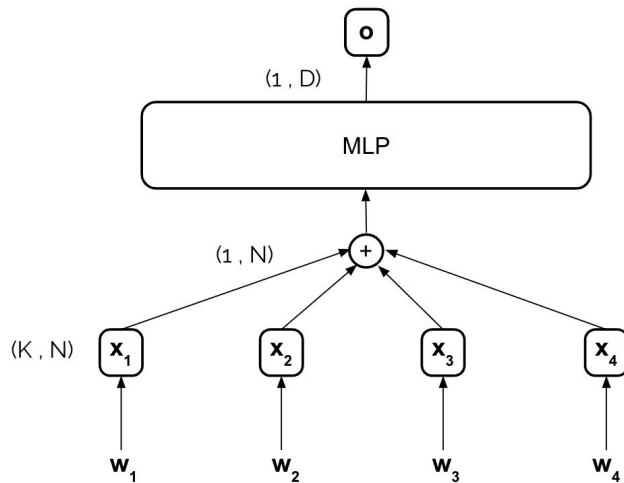
 [Embedding projector](#)

¿Cómo representar texto con embeddings de palabras?



Concatenar

- No escala con respecto al número de tokens.
- Indirectamente, el orden de las palabras importa.



Sumar

- Escala con respecto al número de tokens
- Ignora el orden de las palabras.
- Representaciones extremadamente ruidosas

Modelado de Lenguaje con RNNs

En una RNN, el estado oculto $\mathbf{h}_t$ se actualiza en función del estado oculto anterior $\mathbf{h}_{t-1}$ y la entrada actual. $\mathbf{x}_t$:

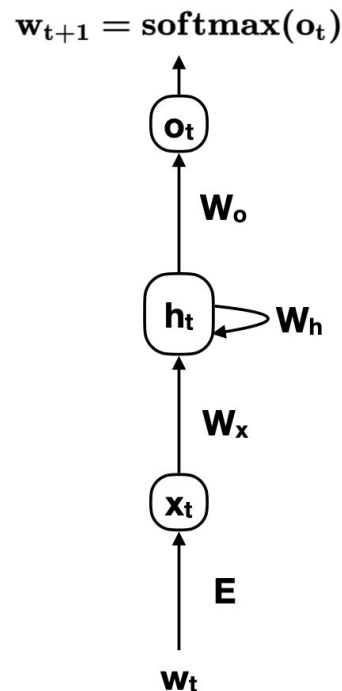
$$\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t)$$

donde $\mathbf{f}$ es una función de activación no lineal (por ejemplo, tanh o ReLU).

La probabilidad de la próxima palabra es::

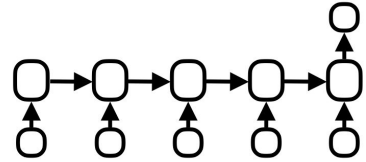
$$P(w_t | w_1, w_2, \dots, w_{t-1}) = \text{softmax}(\mathbf{W}\mathbf{h}_t + \mathbf{b})$$

donde $\mathbf{W}$ y $\mathbf{b}$ son matrices de pesos y biases aprendidos durante el entrenamiento



Aplicaciones de NLP con RNNs

- **Clasificación**
 - Categorización de Texto
 - Named Entity Recognition



Clasificador basado en RNN en Torch

```
import torch
import torch.nn as nn

class TextRNNClassifier(nn.Module):
    def __init__(self, vocab_size, input_size, hidden_size, output_size,
                  n_layers=2, n_classes=2):
        super(TextRNNClassifier, self).__init__()
        self.Emb = nn.Embedding(vocab_size, input_size)
        self.RNN_f = RNN(input_size, hidden_size, output_size, n_layers)
        self.RNN_b = RNN(input_size, hidden_size, output_size, n_layers)
        self.classifier = nn.Linear(hidden_size, n_classes)

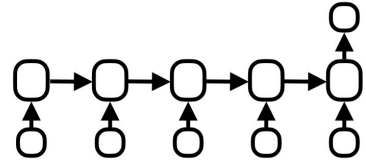
    def forward(self, input):
        word_embeddings = self.Emb(input)
        outputs_f = self.RNN(word_embeddings)
        outputs_b = self.RNN(torch.flip(word_embeddings, dim=(1,)))
        outputs = torch.cat((outputs_f[:, -1, -1, :], outputs_b[:, -1, -1, :]), dim=1)
        logits = self.classifier(outputs.squeeze(1))

    return outputs
```

NLP Applications with RNNs

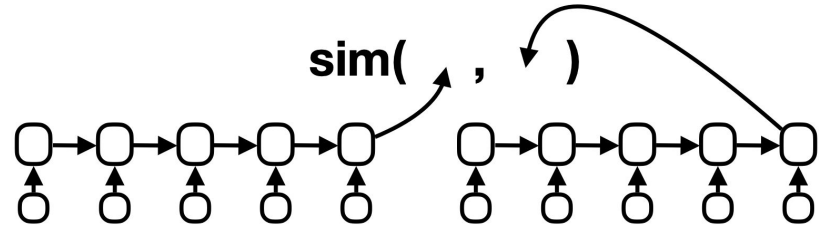
- **Clasificación**

- Categorización de Texto
- Named Entity Recognition



- **Representación de secuencias**

- Similitud de oraciones



Similitud basada en RNN en Torch

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class TextRNNSimilarity(nn.Module):
    def __init__(self, vocab_size, input_size, hidden_size, output_size,
                  n_layers=2):
        super(TextRNNSimilarity, self).__init__()
        self.rnn_encoder = TextRNNEncoder(vocab_size, input_size, hidden_size,
                                           output_size, n_layers)

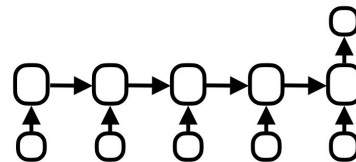
    def forward(self, input_s, input_t):
        output_s = self.rnn_encoder(input_s)[: , -1, -1, :]
        output_t = self.rnn_encoder(input_t)[: , -1, -1, :]
        sim_score = F.cosine_similarity(input_s, input_t)

        return sim_score
```

NLP Applications with RNNs

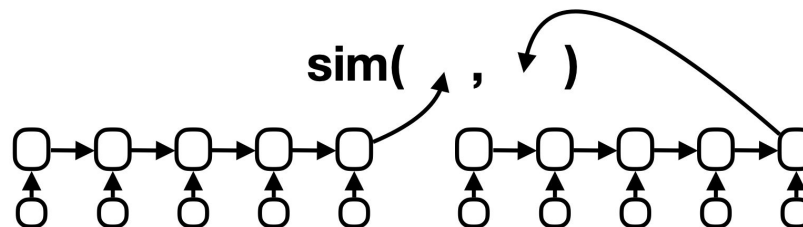
- **Clasificación**

- Categorización de Texto
- Named Entity Recognition



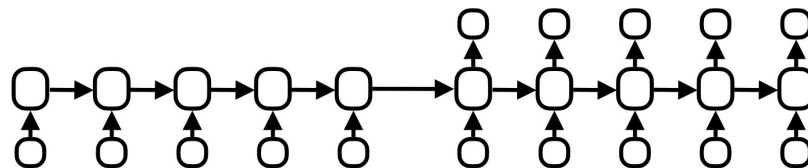
- **Representación de secuencias**

- Similitud de oraciones

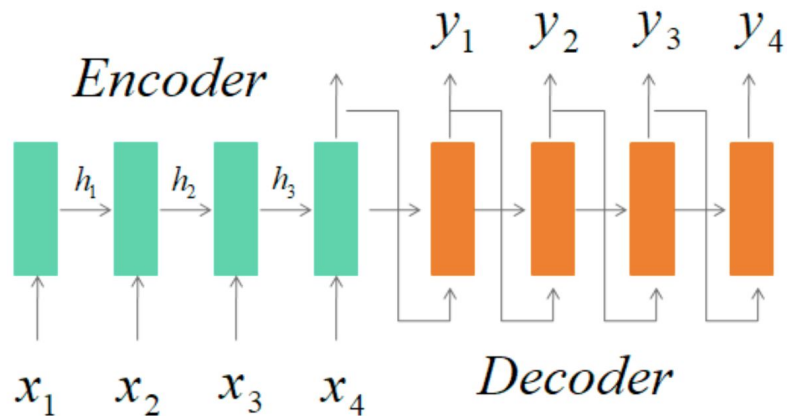
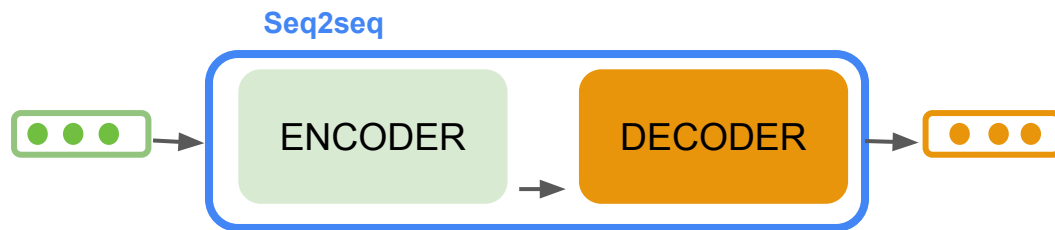


- **Generación de secuencias**

- Traducción automática



Modelos de Secuencia a Secuencia (Seq2seq)



Generador basado en RNN en Torch

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class TextRNNGenerator(nn.Module):
    def __init__(self, vocab_size, input_size, hidden_size, output_size,
                  n_layers=2):
        super(TextRNNGenerator, self).__init__()
        self.rnn_encoder = TextRNNEncoder(vocab_size, input_size, hidden_size,
                                           output_size, n_layers)
        self.rnn_decoder = TextRNNEncoder(vocab_size, input_size, hidden_size,
                                           output_size, n_layers)

        self.word_decoder = nn.Linear(hidden_size, vocab_size)

    def forward(self, input):
        encoded_outputs = self.rnn_encoder(input)[: , -1, -1, :]
        decoder_inputs = torch.zeros(1, 1)
        for _ in range(self.max_gen_length):
            decoder_outputs = self.rnn_decoder(decoder_inputs, hidden=encoded_outputs)[: , -1, -1, :]
            predicted_logits = self.word_decoder(decoder_outputs)
            predicted_words = predicted_logits.argmax(1).unsqueeze(0)
            decoder_inputs = torch.cat((decoder_inputs, predicted_words), dim=1)

        return decoder_inputs
```

Limitaciones de los Modelos de Lenguaje RNN

Limitaciones de los Modelos de Lenguaje
N-gram

QUEREMOS DEPENDENCIAS MÁS LARGAS

Limitaciones de los Modelos de Lenguaje RNN

Limitaciones de los Modelos de Lenguaje
N-gram

WE WANT LONGER DEPENDENCIES

¿Hemos resuelto el problema?

SÍ Y NO

Limitaciones de los Modelos de Lenguaje RNN

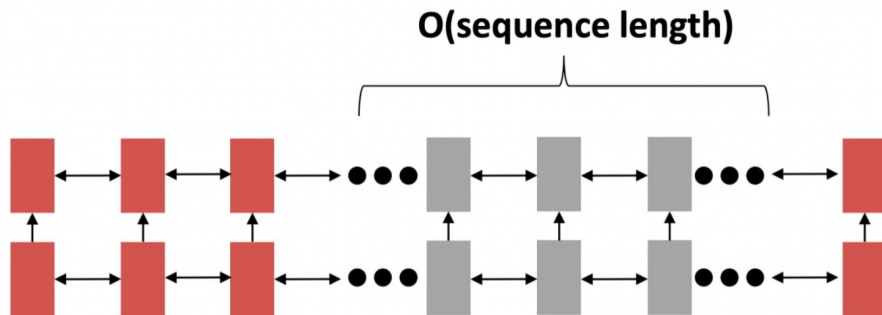
Limitaciones de los Modelos de Lenguaje
N-gram

QUEREMOS DEPENDENCIAS MÁS LARGAS

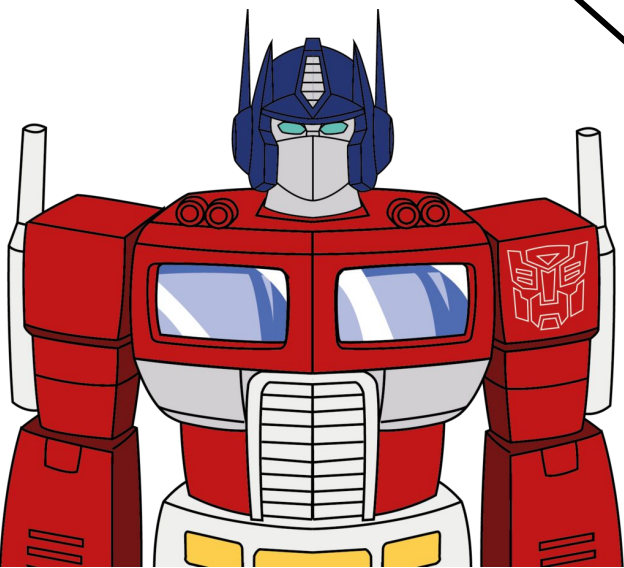
¿Hemos resuelto el problema?

SÍ Y NO

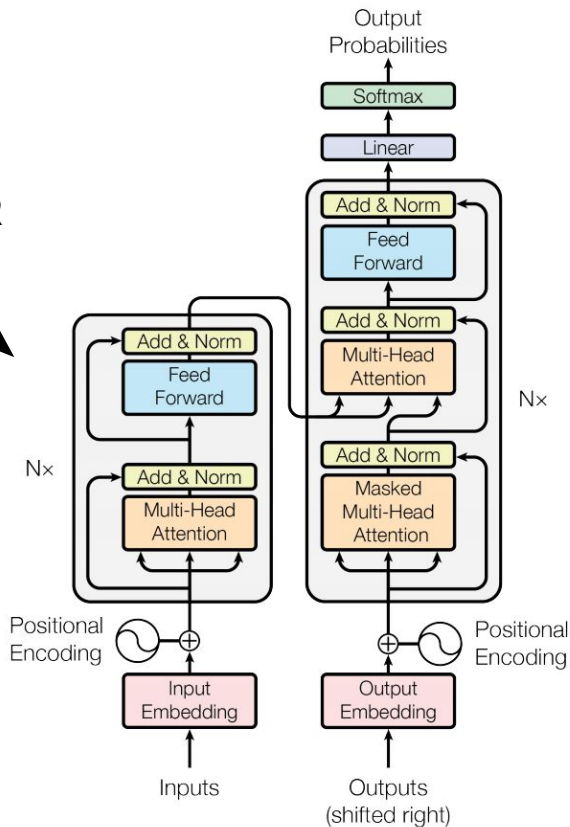
Las relaciones a largo plazo son difíciles de aprender para las RNNs. El Nth token puede contextualizarse con respecto al primer token a través de una representación muy ruidosa (estado oculto después de N pasos de tiempo).



Transformers



ENCODER



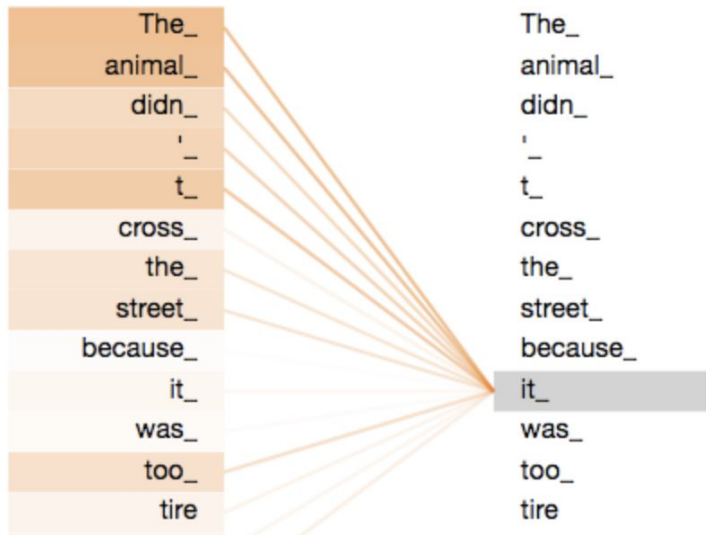
DECODER

Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* (2017).

Self-Attention / Auto-Atención

Los Transformers utilizan la auto-atención para **manejar dependencias a largo plazo**.

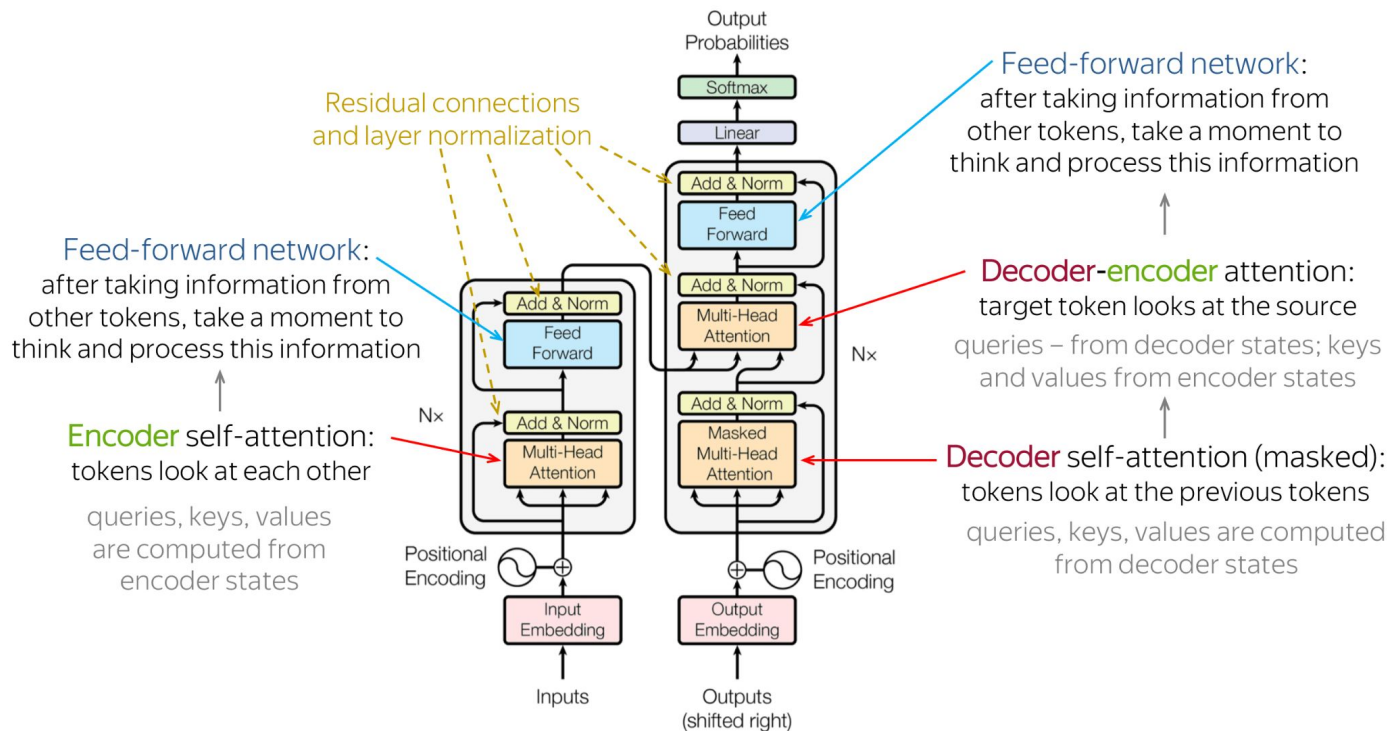
"The animal didn't cross the street because **it** was too tired"



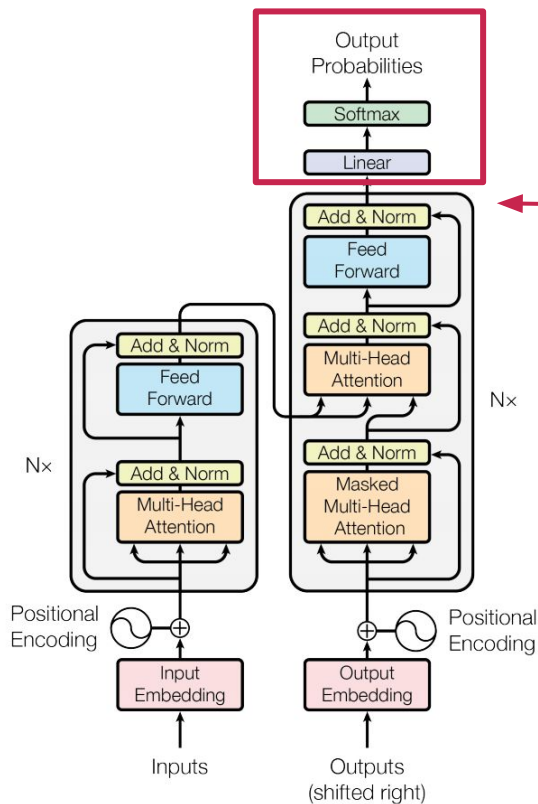
El mecanismo de auto-atención calcula una suma ponderada de los embeddings en la oración.

$$Attention(Q,K,V) = \text{soft max} \left(\frac{QK^T}{\sqrt{d_k}} \right) V$$

Transformers



Transformers



Which word in our vocabulary is associated with this index?

Get the index of the cell with the highest value (argmax)

am

5

log_probs



↑

Softmax

logits



↑

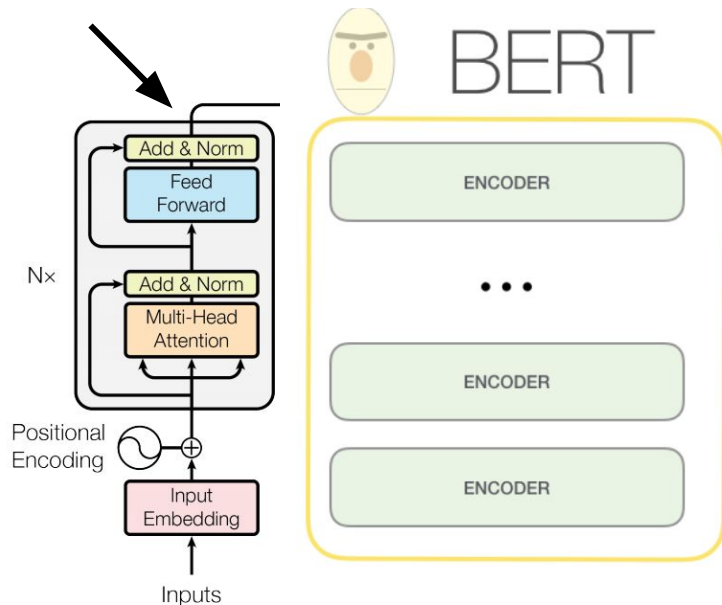
Linear

Decoder stack output



Transformers para el Modelado de Lenguaje

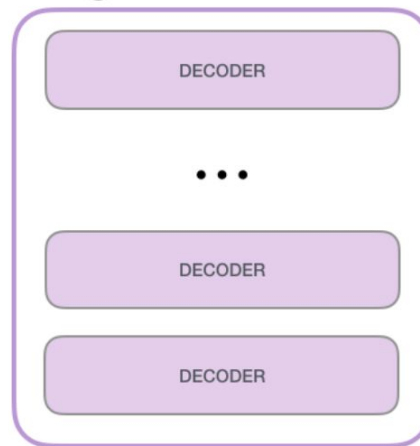
ENCODER



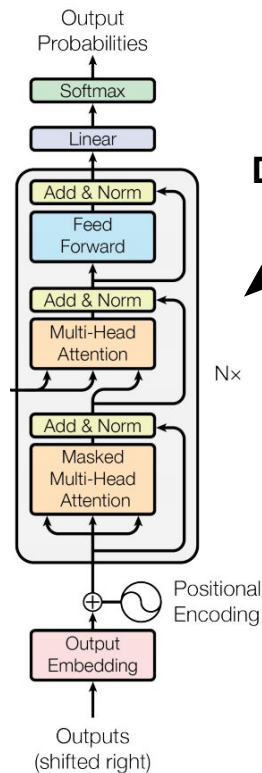
BERT



GPT-2

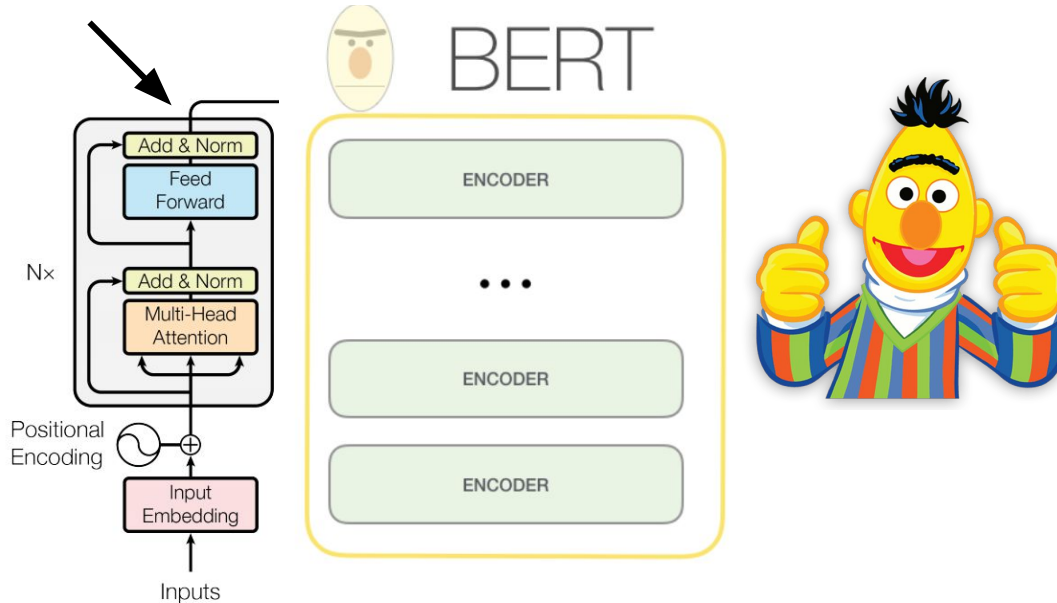


DECODER

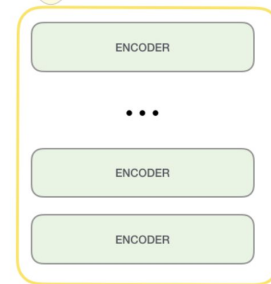


Transformer Bidireccional - BERT

ENCODER



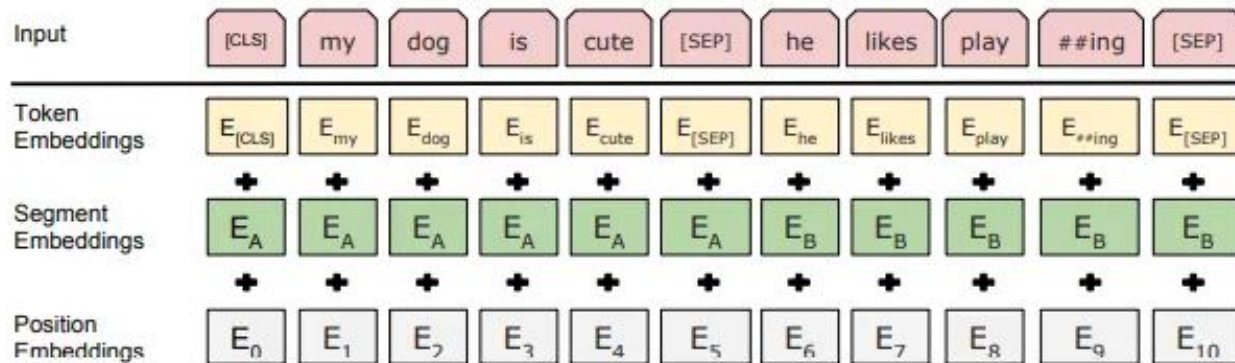
Bidirectional Encoder Representations from Transformers (BERT)



[CLS] sirve para tareas de clasificación y tiene como objetivo resumir la oración en la salida de la última capa de este token.

Cada **token** tiene **acceso directo (attention)** a todos los tokens

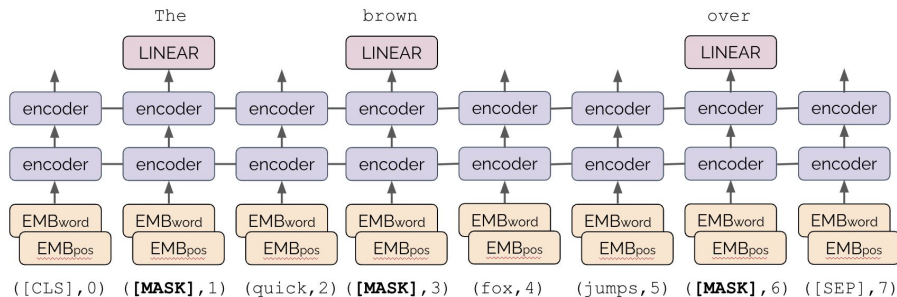
El **[SEP]** token indica que la siguiente oración o segmento empieza.



BERT

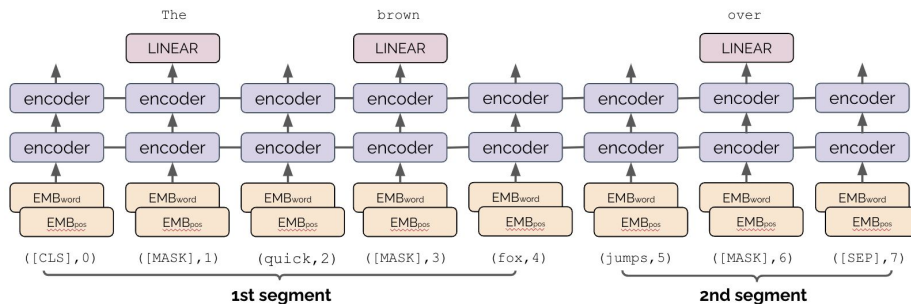
Masked Language Modelling (MLM):

- Enmascarar un subconjunto (15%) de **tokens** e intentar predecirlos.

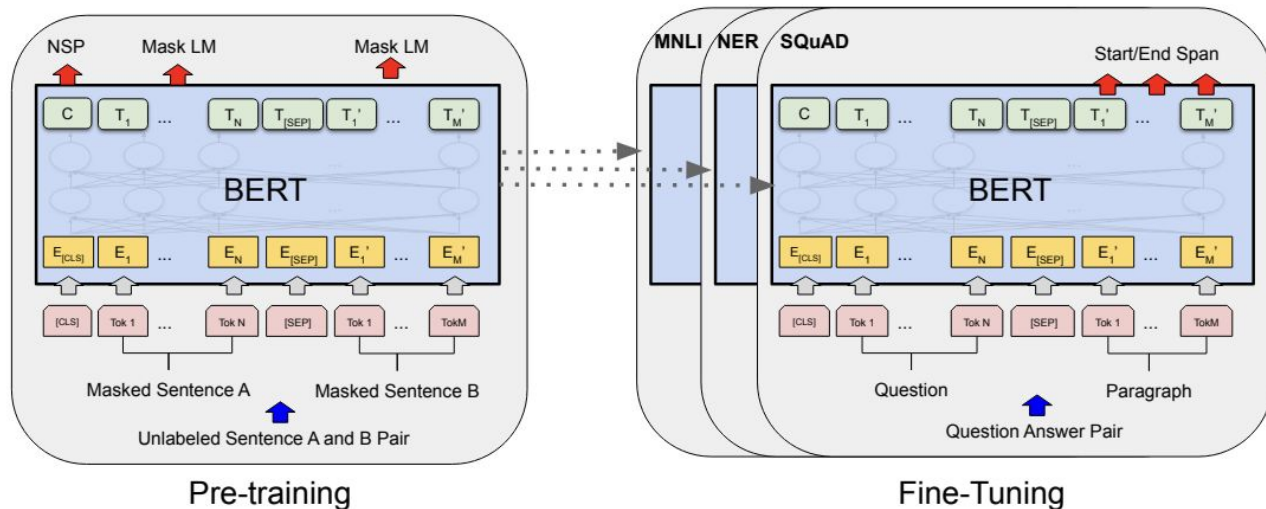


Next Sentence Prediction (NSP):

- Proporcionar un **par de oraciones** y **predecir si la segunda oración sigue a la primera** en el documento original.



Fine-tuning BERT en tareas específicas



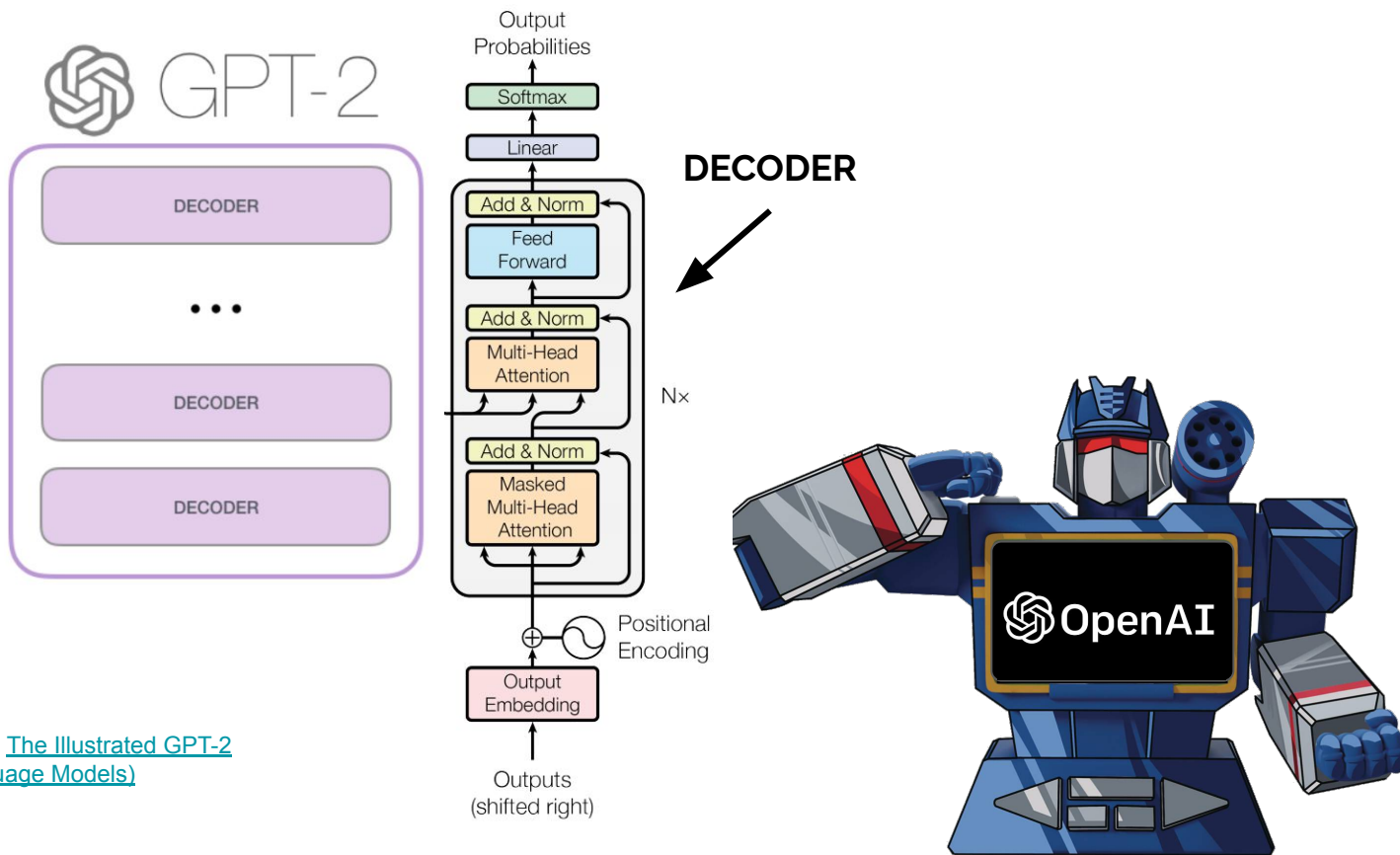
Pre-training:

- Artículos de **Wikipedia** en inglés
- Toronto **Books** Corpus (TBC)

Fine-tuning:

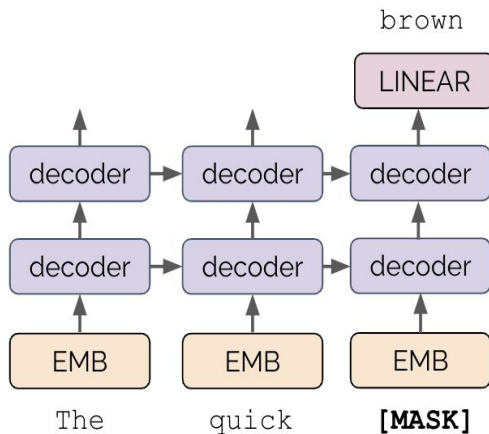
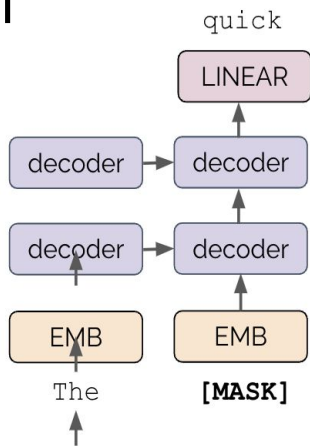
- Conjunto de datos específico de la tarea
- Un modelo para cada tarea

Generative Pre-Training - GPT

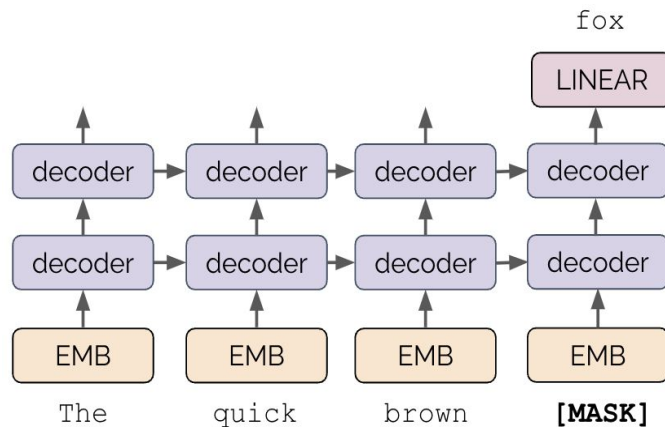


Illustrations from Jay Alammar, [The Illustrated GPT-2](#)
([Visualizing Transformer Language Models](#))

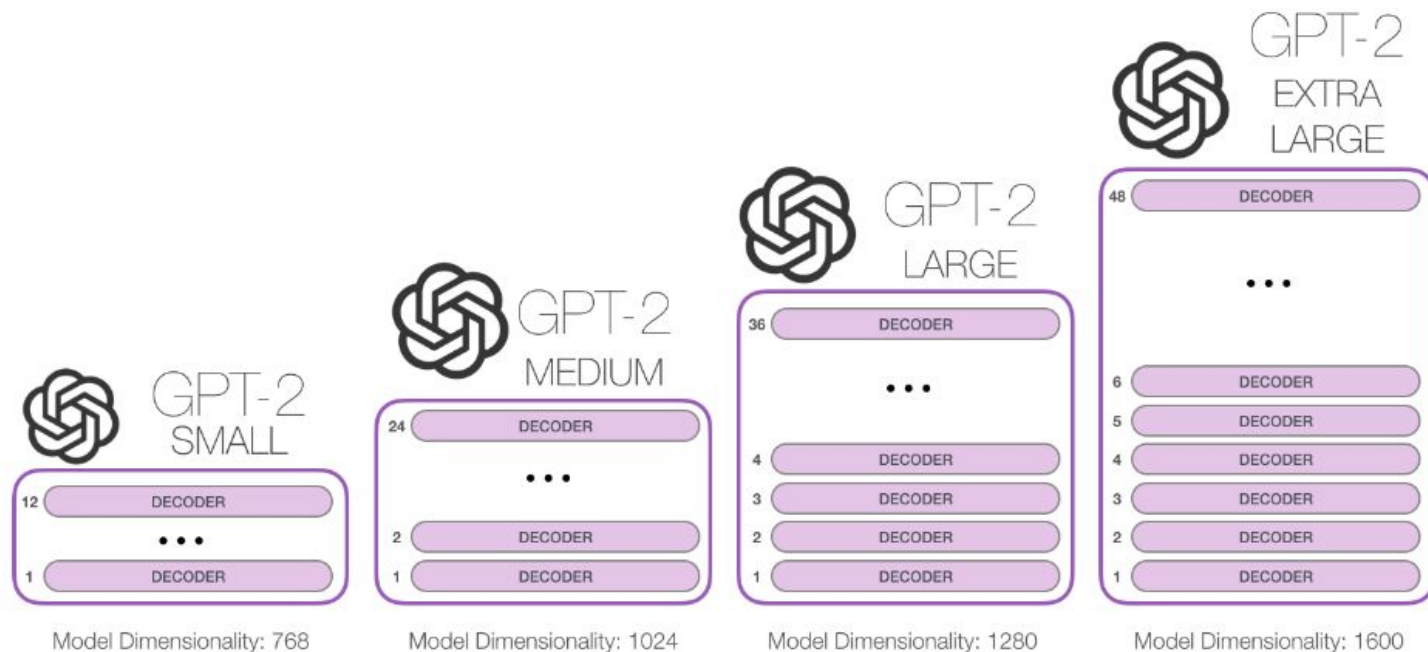
GPT



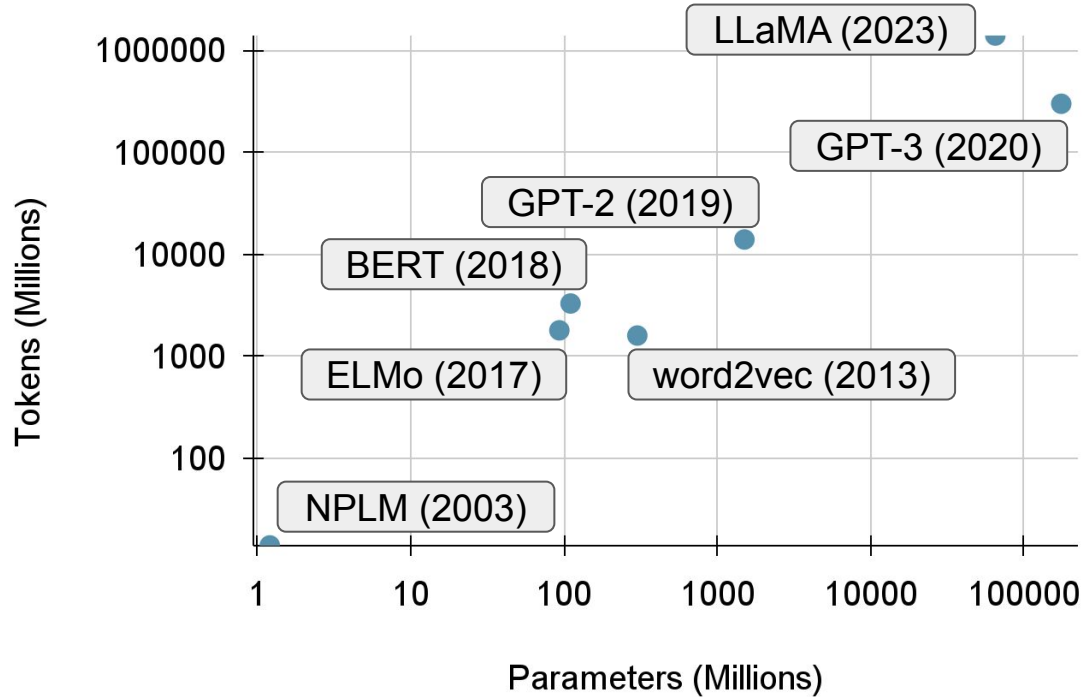
117 M de parámetros, Entrenado en un conjunto de datos de 40 GB (WebText)



GPT-2: Modelos de Lenguaje a Gran Escala



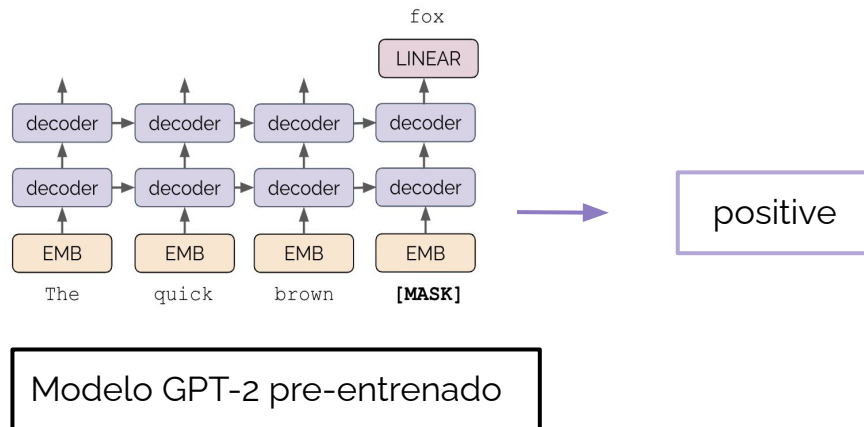
Modelos de Lenguaje a Gran Escala



Generación con Zero-shot Prompting

¡Cualquier tarea dada es una tarea de secuencia a secuencia (generativa)!

Movie Review: "Yesterday, I watch Jaws. It was quite interesting, I think it definitely worth-watching!"
Is this review (a) positive or (b) negative? _____



Zero-Shot con GPT-2

```
from transformers import GPT2LMHeadModel, GPT2Tokenizer

# Load pre-trained GPT-2 model and tokenizer
model_name = "gpt2"
model = GPT2LMHeadModel.from_pretrained(model_name)
tokenizer = GPT2Tokenizer.from_pretrained(model_name)

# Define a prompt for zero-shot learning
prompt = "Translate the following sentence to French: 'The cat is on the  
roof.'\nTranslation:"

# Tokenize the input prompt
inputs = tokenizer(prompt, return_tensors="pt")

# Generate the response
outputs = model.generate(**inputs, max_length=50, num_return_sequences=1)

# Decode the generated text
generated_text = tokenizer.decode(outputs[0], skip_special_tokens=True)

print(generated_text)
```

Generación con Zero-shot Prompting

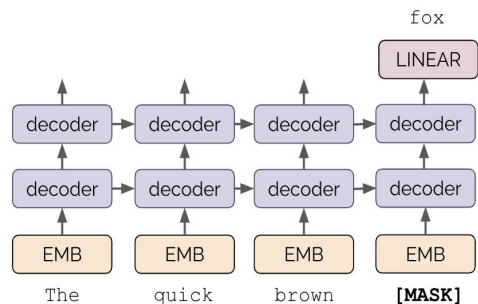
Variabilidad en los Prompts:

La elección del prompt es muy importante para configuraciones de zero-shot.

Movie Review: "Yesterday, I watch Jaws. It was quite interesting, I think it definitely worth-watching!"

Is this
neg

Review: "Yesterday, I watch Jaws. It was quite interesting, I think it definitely worth-watching!" **Is this positive, negative? // _____**



Pre-trained GPT-2 model

positive

negative

Generación con Few-shot Prompting

Formar "ejemplos de demostración" a partir de pares (x, y)

Movie Review: "The cinematography was stellar; great movie!" Is this review (a) positive or (b) negative? Positive

Movie Review: "The plot was boring and the visuals were subpar." Is this review (a) positive or (b) negative? Negative

Movie Review: "The movie's acting could've been better, but the visuals and directing were top-notch." Is this review (a) positive or (b) negative? Positive

Movie Review: "Yesterday, I watch Jaws. It was quite interesting, I think it definitely worth-watching!" Is this review (a) positive or (b) negative? _____

Few-Shot con GPT-2

```
# Define a prompt with a few-shot learning approach
prompt = (
    "Translate the following sentences to French:\n"
    "1. 'The cat is on the roof.'\n"
    "Translation: 'Le chat est sur le toit.'\n\n"
    "2. 'The dog is in the garden.'\n"
    "Translation: 'Le chien est dans le jardin.'\n\n"
    "3. 'The bird is in the tree.'\n"
    "Translation: 'L'oiseau est dans l'arbre.'\n\n"
    "4. 'The fish is in the pond.'\n"
    "Translation: "
)

# Tokenize the input prompt
inputs = tokenizer(prompt, return_tensors="pt")

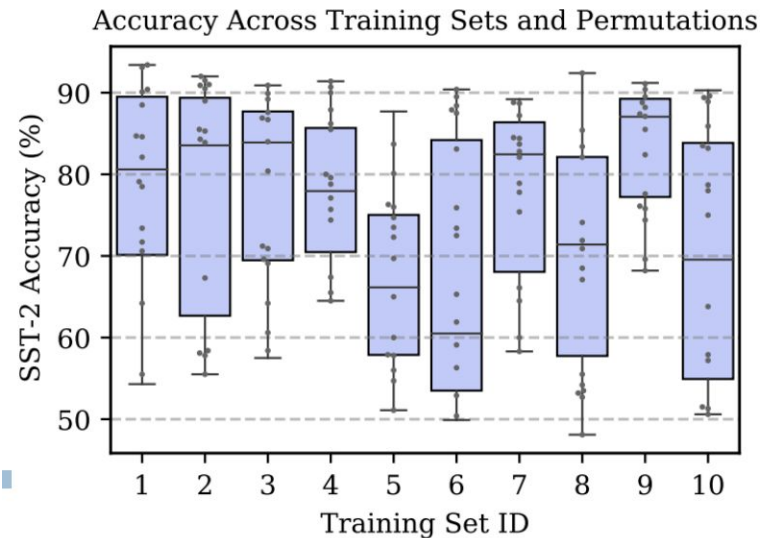
# Generate the response
outputs = model.generate(*inputs, max_length=100, num_return_sequences=1)

# Decode the generated text
generated_text = tokenizer.decode(outputs[0], skip_special_tokens=True)

print(generated_text)
```

Few-shot Prompting

- El Few-shot Prompting no requiere actualizaciones de parámetros y se realiza directamente en LLMs pre-entrenados.
- El rendimiento de few-shot prompting depende en gran medida de las características de demostración, incluyendo el formato, el orden de los ejemplos, entre otros.



Instruction-based fine-tuning

Queremos optimizar los modelos para $P(\text{respuesta}|\text{prompt}, \text{entrada})$, pero solo aprendieron un objetivo básico de modelado de lenguaje (e.g., **predecir la próxima palabra**)

Podríamos tratar el modelado básico de lenguaje como pre-entrenamiento (pre-training) y luego ajustarlos (fine-tuning) en lo que nos importa.

Pre-training → **Fine-tuning** en conjuntos de datos específicos de la tarea.

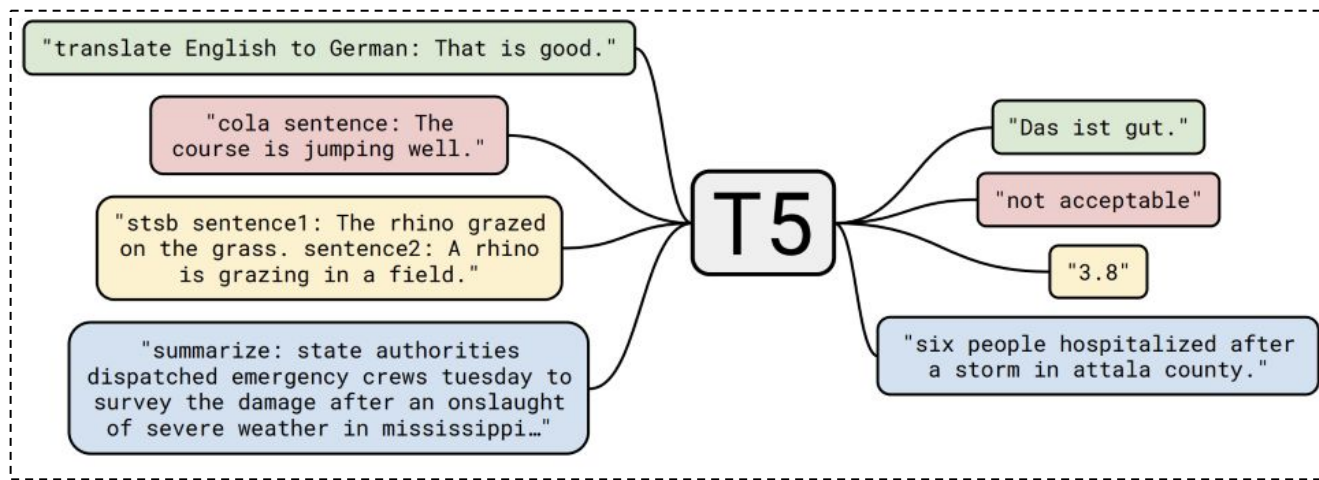
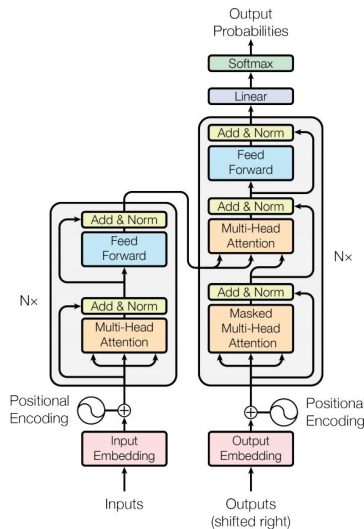
Instruction tuning: fine-tuning supervisado en conjuntos de datos de muchas tareas de NLP descritas por instrucciones.

Caso: T5 como modelo de lenguaje base + To para instruction tuning

T5's Text-to-Text Framework

Encoder-Decoder

Idea principal: **convertir** todas las **tareas** de NLP en un **formato de texto a texto**.



T5 como modelo de lenguaje pre-entrenado

Pre-training: T5 está pre-entrenado en un gran corpus usando un objetivo de “corrupción de fragmentos” (“**span corruption**” **objective**) .

- Corromper secuencias de entrada y entrenar el modelo para reconstruir el texto original.

T5

Original text

Thank you ~~for inviting~~ me to your party ~~last~~ week.

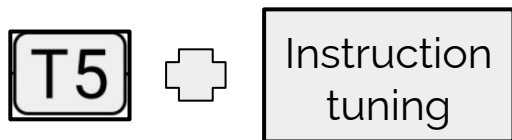
Inputs

Thank you <X> me to your party <Y> week.

Targets

<X> for inviting <Y> last <Z>

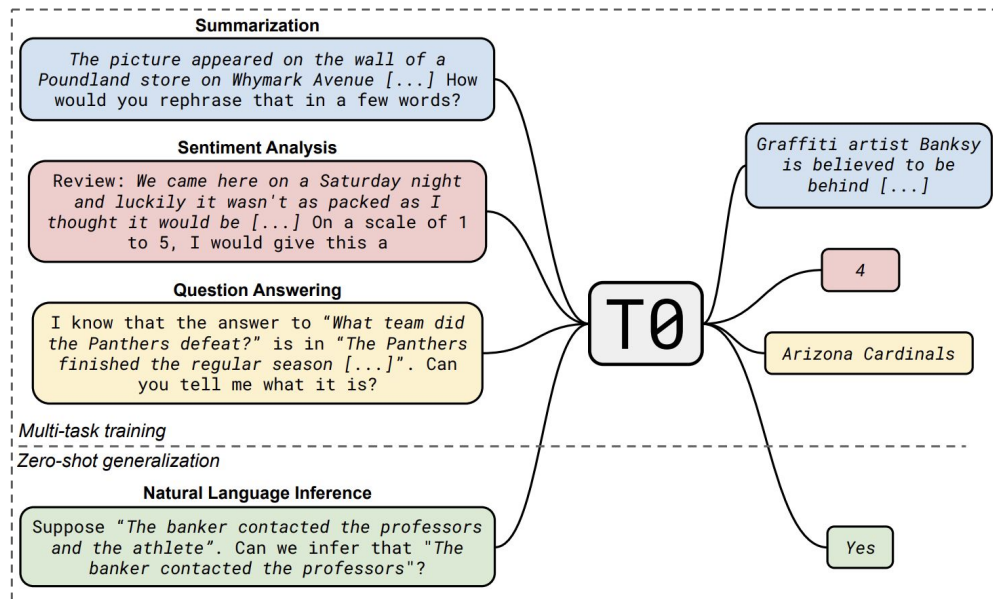
To - Instruction Tuning



T0 está entrenado en una variedad de tareas, cada una formateada como una tarea de texto a texto.

Los prompts son descripciones en lenguaje natural que explican la tarea.

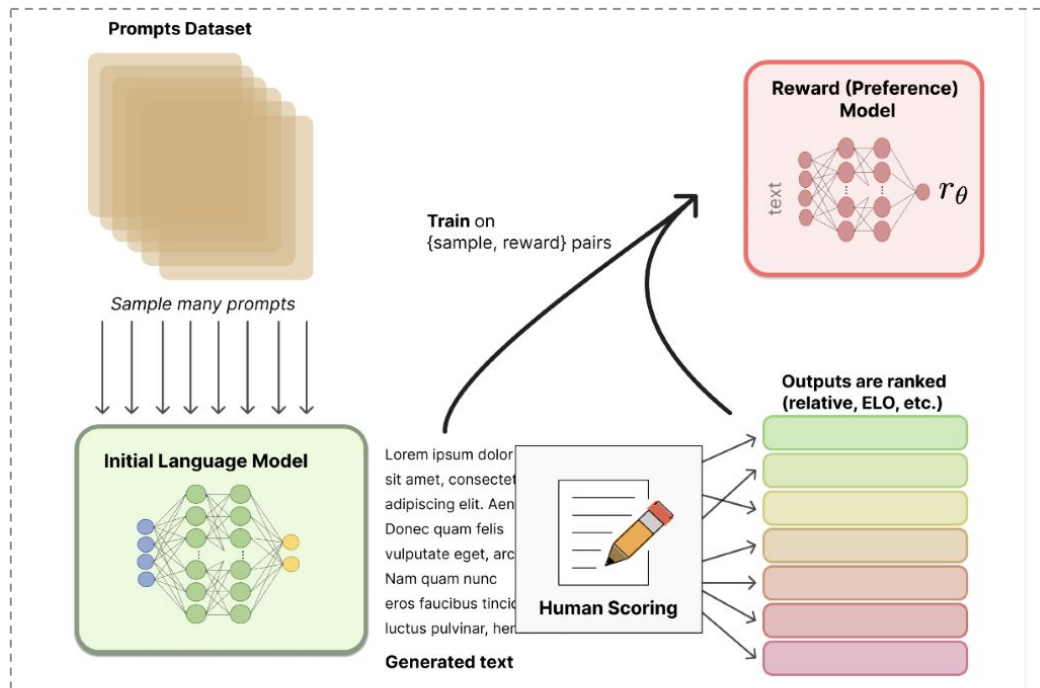
T0 se basa en la arquitectura **T5**: misma arquitectura basada en transformers con **encoder-decoder** layers.



Reinforcement Learning from Human Feedback (RLHF)

Alignment with Human Feedback - RLHF

- Alinear las respuestas de los modelos con los deseos humanos conduce a mejores respuestas.



Reinforcement Learning from Human Feedback (RLHF)

Pre-training

Step 0.1

Collect large text corpora and train a GPT model.

A document is sampled from our corpus.

Barack Obama
was born in
August 4, 1961.

The document is used to train a GPT model with self-supervision.



Barack Obama
was born ...

Instruction-tuning

Step 0.2

Fine-tune the pre-trained GPT model on instruction-based datasets.

A document is sampled from the datasets.

When was Barack
Obama born?
In August 4, 1961.

The document is used to train a GPT model with self-supervision.



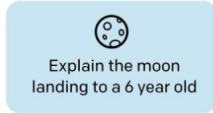
When was Barack
Obama born?

Reinforcement Learning from Human Feedback (RLHF)

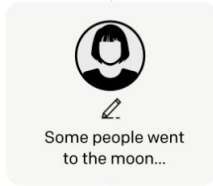
Step 1

**Collect demonstration data,
and train a supervised policy.**

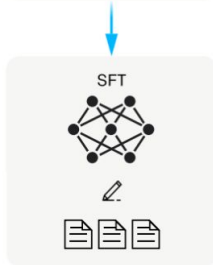
A prompt is
sampled from our
prompt dataset.



A labeler
demonstrates the
desired output
behavior.



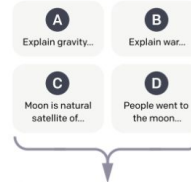
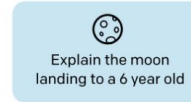
This data is used
to fine-tune GPT-3
with supervised
learning.



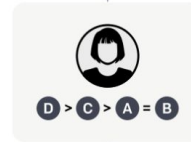
Step 2

**Collect comparison data,
and train a reward model.**

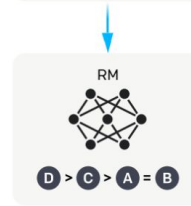
A prompt and
several model
outputs are
sampled.



A labeler ranks
the outputs from
best to worst.



This data is used
to train our
reward model.



Step 3

**Optimize a policy against
the reward model using
reinforcement learning.**

A new prompt
is sampled from
the dataset.

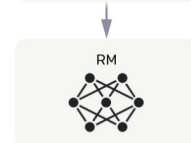


The policy
generates
an output.

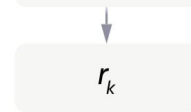


Once upon a time...

The reward model
calculates a
reward for
the output.



The reward is
used to update
the policy
using PPO.



Reinforcement Learning from Human Feedback (RLHF)

Table 1: Distribution of use case categories from our API prompt dataset.

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

Table 2: Illustrative prompts from our API prompt dataset. These are fictional examples inspired by real usage—see more examples in Appendix A.2.1.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: "" {summary} "" This is the outline of the commercial for that play: ""

Reinforcement Learning from Human Feedback (RLHF)

RLHF es un factor clave en muchos modelos populares, incluyendo ChatGPT, Bard/Gemini, Claude, Llama 2 y más...

Reinforcement Learning from Human Feedback (RLHF)

RLHF es un factor clave en m...
Claude, Llama 2 y más...

ChatGPT, Bard/Gemini,

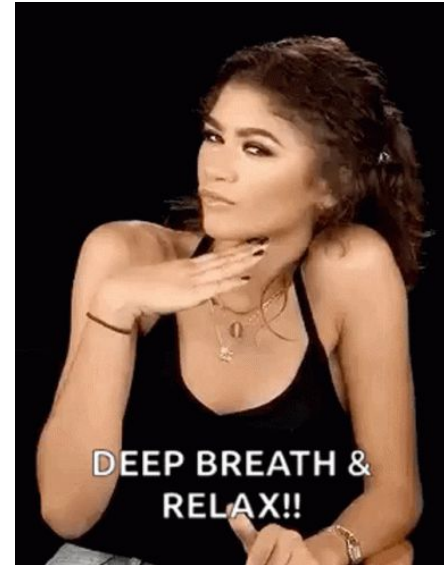


2. Limitaciones de los Modelos de Lenguaje de gran tamaño

Limitaciones de los LLMs

Entender las Limitaciones es Crucial

- Asegurar expectativas realistas
- Identificar áreas de mejora
- Mitigar riesgos y preocupaciones éticas



Ambigüedad y Malinterpretación

Los LLMs tienen dificultades con consultas ambiguas y lenguaje matizado.



Image generated with GPT-4o

Puntos de Interés (POI) en las Redes Sociales

Clasificación multiclase a nivel de publicaciones en redes sociales.



imagine all the people
sharing all the world ~

Arts & Entertainment

College & University

Great Outdoors

Nightlife Spot

Professional

Shop & Service

Travel & Transport

Points-of-Interest (POI) in Social Media

Prompt:

What type of place was the next tweet sent from? Select one of the next categories: Arts & Entertainment, College & University, Food, Great Outdoors, Nightlife Spot, Professional & Other Places, Shop & Service, Travel & Transport. Answer with the category name directly. Tweet: "**imagine all the people sharing all the world ~**"

imagine all the people
sharing all the world ~

- Arts & Entertainment
- College & University
- Great Outdoors
- Nightlife Spot
- Professional
- Shop & Service
- Travel & Transport

What type of place was the next tweet sent from? Select one of the next categories: Arts & Entertainment, College & University, Food, Great Outdoors, Nightlife Spot, Professional & Other Places, Shop & Service, Travel & Transport. Answer with the category name directly. Tweet: "imagine all the people sharing all the world ~"

Model A: qwen1.5-110b-chat

*The tweet does not provide specific information to determine the exact type of place it was sent from. However, given the content of the tweet, which is a reference to the lyrics from John Lennon's song "Imagine," it doesn't directly align with any of the provided categories in a literal sense. If forced to choose based on the sentiment rather than a literal location, one might consider "**Arts & Entertainment**" since the tweet quotes a song, but this is an interpretation and not a factual categorization based on the tweet's metadata.*

Model B: GPT-4o

Arts & Entertainment

What type of place was the next tweet sent from? Select one of the next categories: Arts & Entertainment, College & University, Food, Great Outdoors, Nightlife Spot, Professional & Other Places, Shop & Service, Travel & Transport. Answer with the category name directly. Tweet: "imagine all the people sharing all the world ~"



imagine all the people
sharing all the world ~

Model B: GPT-4o

Great Outdoors

What type of place was the next tweet sent from? Select one of the next categories: Arts & Entertainment, College & University, Food, Great Outdoors, Nightlife Spot, Professional & Other Places, Shop & Service, Travel & Transport. Answer with the category name directly. Tweet: "imagine all the people sharing all the world ~"



Model B: GPT-4o

Great Outdoors

La multimodalidad reduce la ambigüedad.



imagine all the people
sharing all the world ~

Modelos Multimodales

Los modelos multimodales procesan conjuntamente información de dos o más modalidades de entrada, por ejemplo, imágenes y texto, voz y video, etc.



Image generated with GPT-4o

Grados de Multimodalidad

Las plataformas de redes sociales a menudo forman "cámaras de eco" que alientan a los usuarios a leer solo contenido que confirme creencias que ya tienen (Getty)

Débil



Descripción de la imagen: Una mujer con un traje gris oscuro está dando un discurso

Fuerte

Panofsky. (1939). Studies in Iconology.

<https://www.independent.co.uk/news/angela-merkel-says-internet-search-engines-endangering-debate-algorithms-should-be-revealed-a7383811.html>

Publicaciones Multimodales en Redes Sociales

Combinar información de texto e imagen es un desafío porque la semántica entre modalidades puede estar oculta o la relación entre imagen y texto puede ser débil

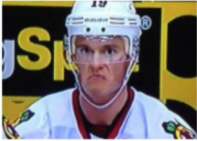
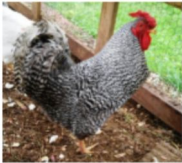
Image	Text (Post)	Image-Text Relation in Post	Image Caption
	When @USER gets more followers than you in 12 hours	The image complements the text to provide meaning of the post	A close up of a hockey player wearing a helmet
	My baby approves	The image does not add to the meaning of the post and the text does not provide a description of the image	A gray and white chicken standing in the dirt

Image-Text Contrastive (ITC)

Image-Text Contrastive (ITC) está diseñada para minimizar la distancia entre las representaciones de imagen y texto dentro de una publicación.

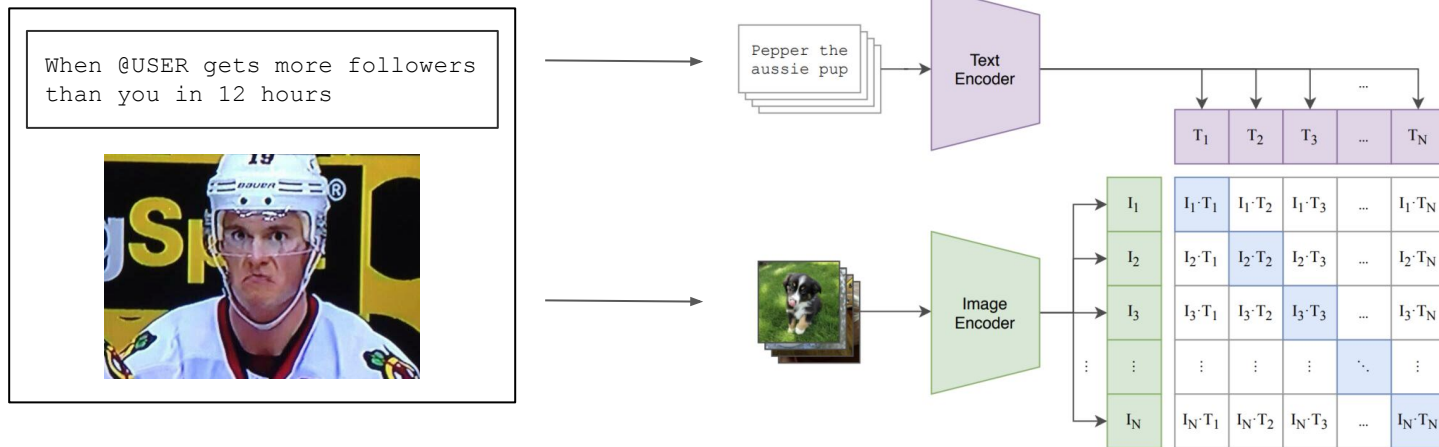
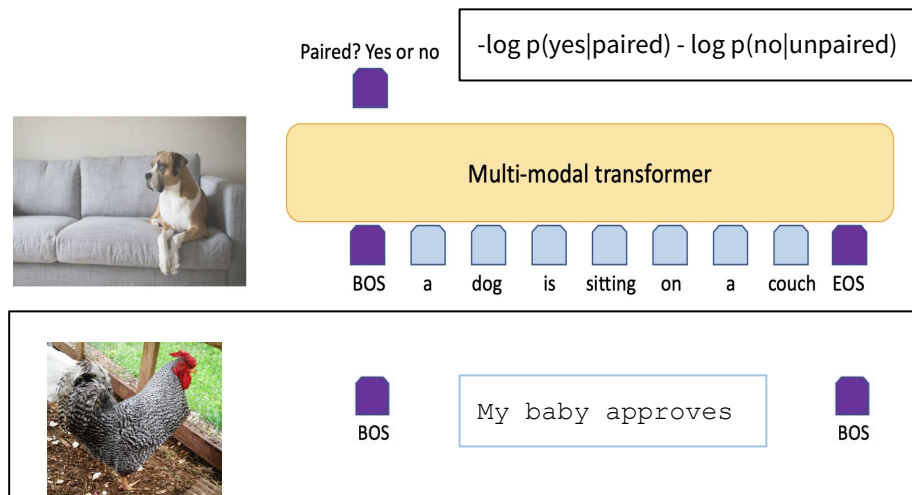


Image-Text Matching (ITM)

Image-Text Matching (ITM) mejora la capacidad del modelo para comprender la relación semántica entre imágenes y texto.



Models

Text-only

- Bert
- Bernice
- Flan-T5 (FS prompt)
- GPT-3 (FS prompt)

Image-Only

- ResNet
- ViT

Text & Image

- Ber-ViT-Conc
- Ber-ViT-Att
- MMBT
- LXMERT
- ViLT

Text & Image + AUX

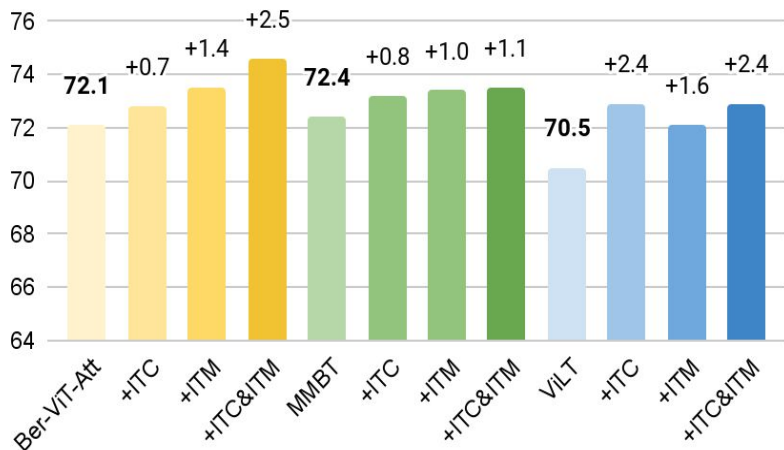
- + ITC
- + ITM
- + ITC & ITM

Combinar
características de
texto e imagen

Entrenar
conjuntamente en
una configuración
multitarea

Resultados

MVSA



- **Multi-View Sentiment Analysis**
- 3 clases

F1

Ambigüedad y Malinterpretación

Los LLMs tienen dificultades con consultas ambiguas y lenguaje matizado.



Image generated with GPT-4o

Ambigüedad y Malinterpretación

Los LLMs tienen dificultades con consultas ambiguas y lenguaje matizado.

La multimodalidad reduce la ambigüedad, pero es esencial gestionar los ejemplos de entrenamiento de imagen y texto y **considerar** las **variaciones en la complejidad de los datos**.



Image generated with GPT-4o

Cobertura de Lenguajes Limitada

NLP generalmente cubre **100 idiomas de los 3,000 idiomas** escritos (400 con más de 1 millón de hablantes).



Image generated with GPT-4o

van Esch et al. Writing System and Speaker Metadata for 2,800+ Language Varieties. LREC 2022.
Joshi et al. The State and Fate of Linguistic Diversity and Inclusion in the NLP World. ACL 2020.

Caso de Uso: Traducción Automática

- “NLLB” (“ningún idioma dejado atrás”) es una familia de modelos de traducción automática publicada por Meta AI en 2022, específicamente diseñada para entornos con pocos recursos.
- Entrenada para traducir una oración entre cualquiera de los **202 idiomas**.



¿Podemos incluir más idiomas?

- Mirar los datos de entrenamiento
- Examinar la tokenización del nuevo idioma
- Actualizar el vocabulario
- Agregar nuevos tokens de idioma
- Entrenar el modelo neuronal
- Evaluar el modelo



¿NLLB incluye idiomas indígenas de las Américas?

✓ Aymara (aym), Guarani (gn) and Quechua (quy)



Yucatec Maya writing in the *Dresden Codex*, ca. 11–12th century, Chichen Itza

¿NLLB incluye idiomas indígenas de las Américas?

- ✓ Aymara (aym), Guaraní (gn) and Quechua (quy)
- ✗ **Bribri (bzd), Asháninka (cni), Chatino (czn), Wixarika (hch), Nahuatl (nah), Hñähñu (oto), Shipibo-Konibo (shp), and Rarámuri (tar)**



Yucatec Maya writing in the *Dresden Codex*, ca. 11–12th century, Chichen Itza

¿Cuál es el rendimiento de NLLB en aym, gn y quy?

Model	quy	aym	gn
Baseline (Vázquez et al., 2021)	33.8	32.7	31.1
Inference			
600M distilled	30.0	34.2	32.5
1.3B distilled	31.0	35.2	35.2
1.3B	31.2	34.5	34.3
3.3B	32.9	35.4	35.6

Valid ChrF

- "Character F-score," mide la similitud entre un texto traducido automáticamente y una traducción de referencia basada en n-gramas de caracteres.
- Es útil para idiomas con morfología rica o formas de palabras complejas.

- La inferencia en aym y gn supera significativamente un modelo base anterior.
- Los resultados en quy son peores
- Los modelos más grandes proporcionan mejores resultados
- El modelo de 3.3B es mucho más lento de entrenar que el de 1.3B.

Incluyendo 8 idiomas indígenas.

Data

- Shared task, Helsinki's 2021 submission, REPUcs, NLLB Seed and NLLB Multi-Domain
- Bible data del JHU Bible corpus
- Backtranslations

Preprocessing

- Extendemos la matriz de embeddings de NLLB para cubrir las 8 etiquetas de idioma adicionales.
- El tokenizador de NLLB cubre la mayoría de los caracteres en los datos de entrenamiento.
- Reemplazamos la puntuación con equivalentes compatibles:
 - « —> <<
 - " " " —> "

Training

- Entrenamos todos los parámetros de los modelos.

Test Set Results

- Nuestro Submission 1 logró el promedio chrF más alto entre todos los idiomas.
- También logramos el chrF más alto para cuatro de los once idiomas individuales.

Submission	aym	bzd	cni	czn	gn	hch	nah	oto	quy	shp	tar	mean
3	35.3	24.5	28.5	39.9	39.1	32.0	27.3	14.8	37.2	28.6	18.4	29.6
2	36.2	24.4			39.3				39.3	33.4		30.3
1		25.0		40.0		32.3			39.5		18.7	30.5

Cobertura de Lenguajes Limitada

NLP generalmente cubre **100 idiomas de los 3,000 idiomas** escritos (400 con más de 1 M de hablantes).

Podemos aumentar la cobertura entrenando con conjuntos de datos de idiomas con pocos recursos



Image generated with GPT-4o

Cobertura de Lenguajes Limitada

NLP generalmente cubre **100 idiomas** de los **7.000 idiomas** escritos (400 con más de 1 M de hablantes).

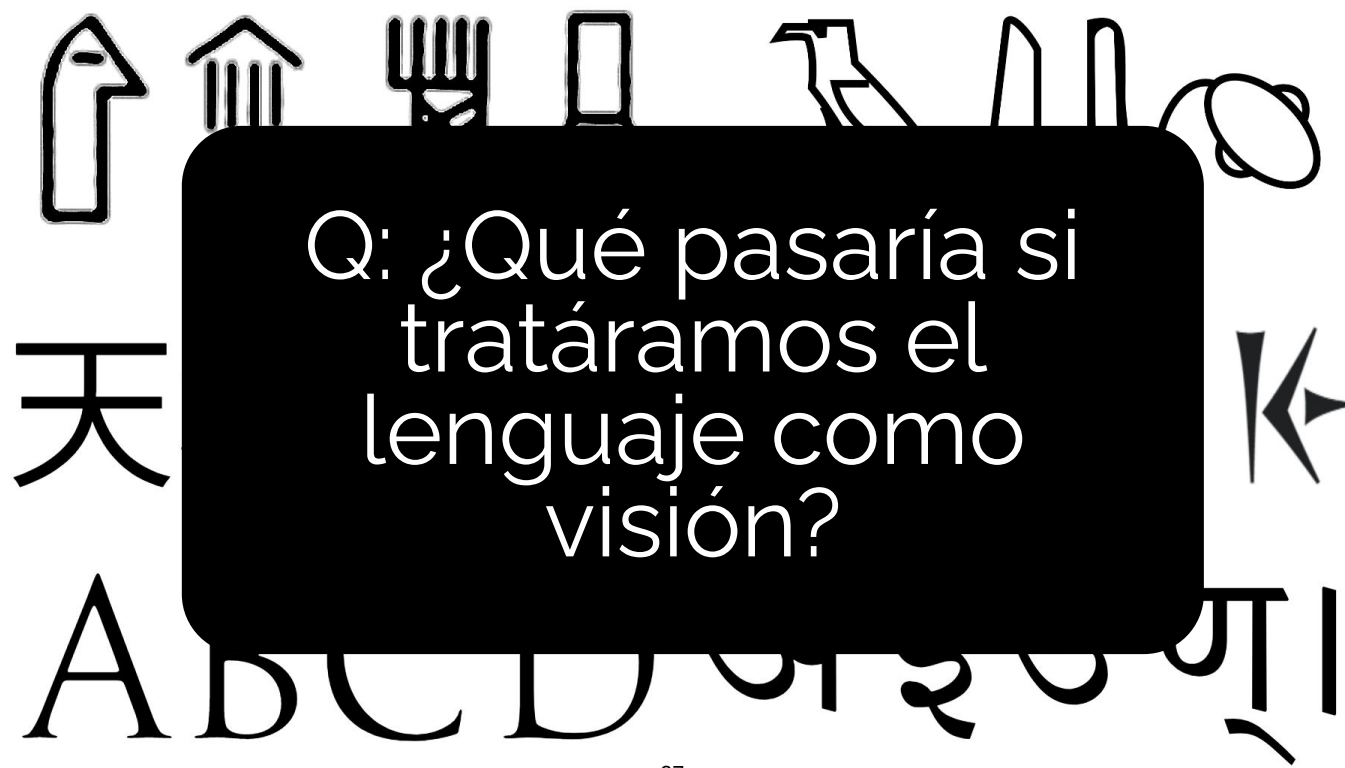
Podemos aumentar la cobertura

idiomas con pocos recursos

¿Podemos hacer algo mejor?

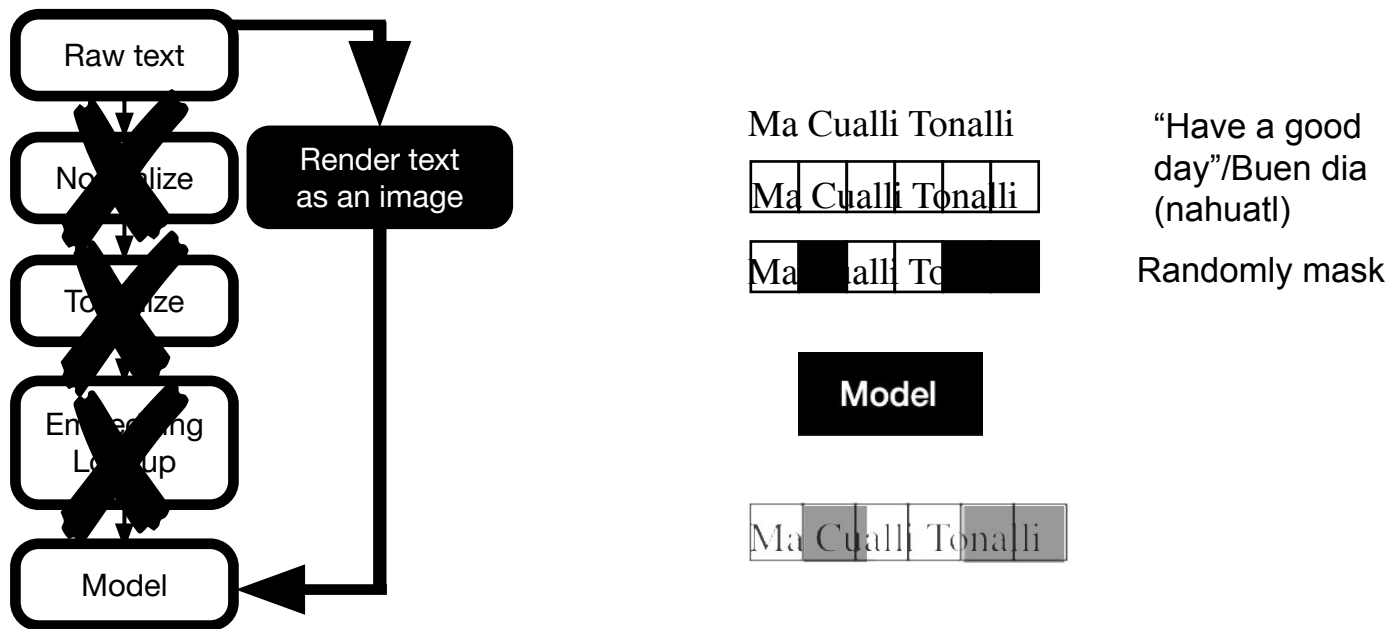


Image generated with GPT-4o

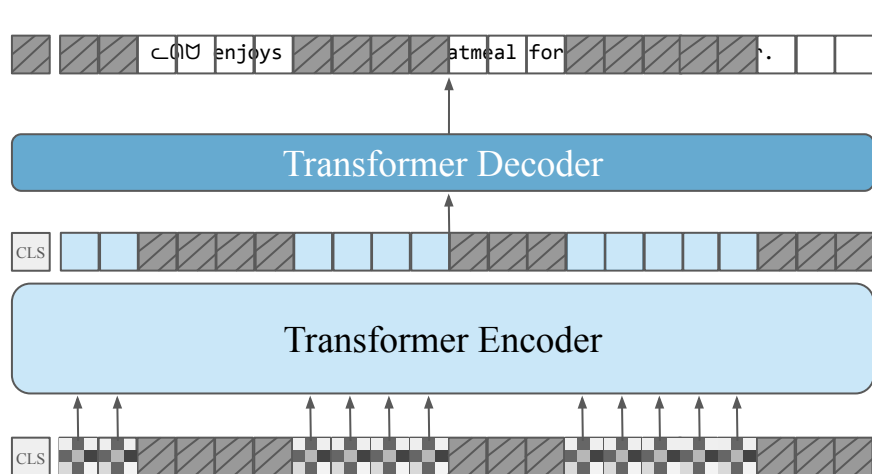


Q: ¿Qué pasaría si
tratáramos el
lenguaje como
visión?

Tratando el lenguaje como visión



El Modelo PIXEL



$$\frac{1}{M} \sum_{m \in M} (Y_m - \hat{Y}_m | \mathbf{x}_{\setminus m})^2$$

8 Layers

12 Layers

16pixel x 16pixel patch

Google Noto Fonts

PyGame / PangoCairo

3 CLS Embedding & Span Mask m patches

2 Projection + Position Embedding

1 Render Text as Image



My cat enjoys eating warm oatmeal for lunch and dinner.

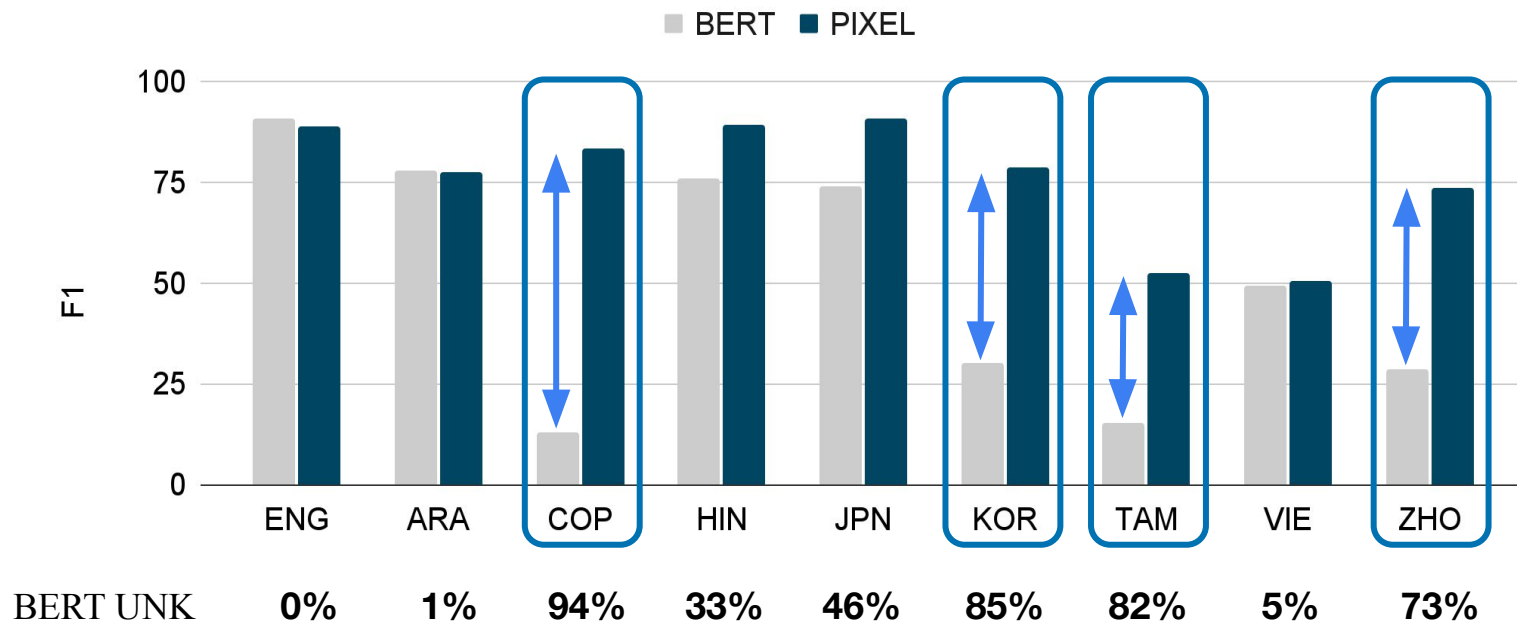
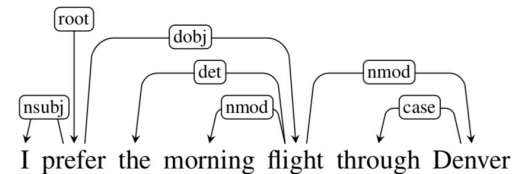
Pre-entrenamiento

- **Conjunto de Datos en Inglés:** Wikipedia en Inglés y Corpus de Libros
- **Enmascaramiento:** Enmascaramiento de 25% de fragmentos
- **Longitud máxima de secuencia:** 529 parches (16x8464 pixeles)
- 8 GPUs A100 de 40GB durante 8 días
- **Parámetros:** 86M encoder + 26M decoder

Solo hay un 0.05% de texto no inglés en nuestros datos de preentrenamiento (estimado por Blevins y Zettlemoyer 2022).

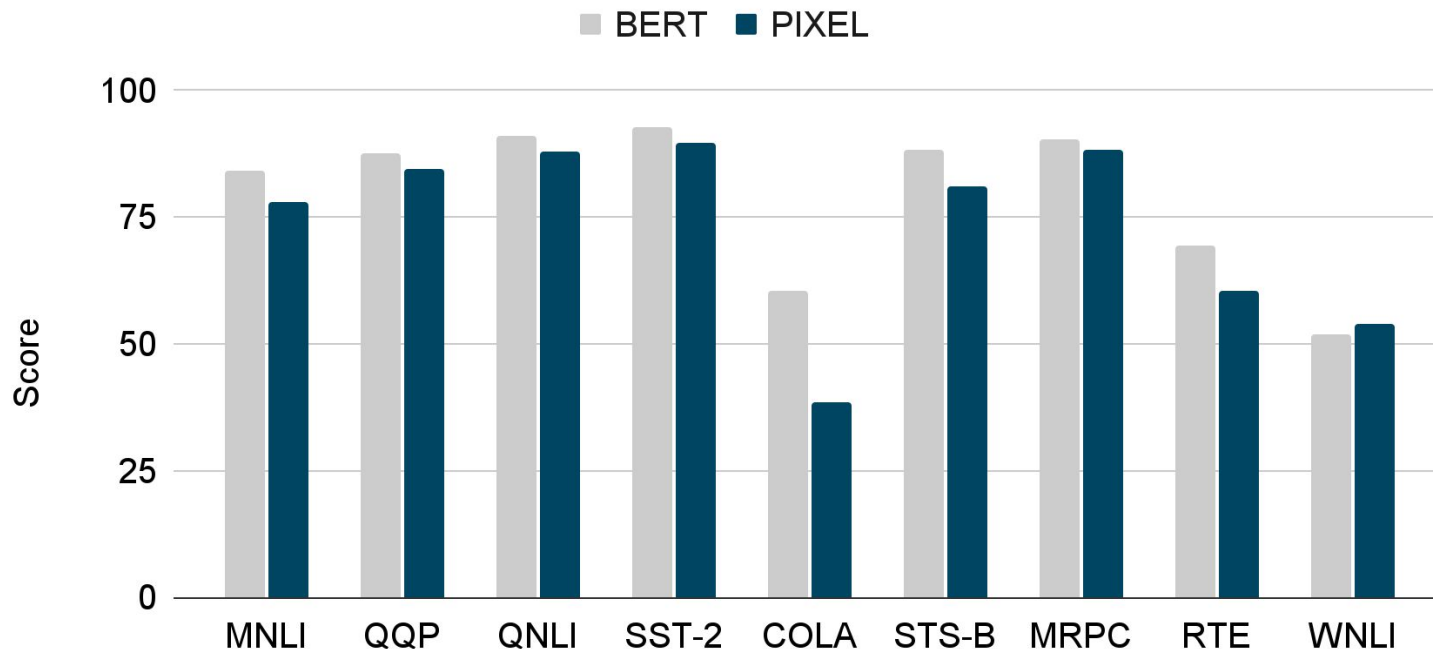
The Great Wall of China (traditional Chinese: 萬里長城; simplified Chinese: 万里长城; pinyin: Wànlǐ Chángchéng)

Resultados: Dependency Parsing



PIXEL supera enormemente a BERT en scripts no vistos

GLUE: Comprensión a Nivel de Oración



BERT supera a PIXEL en tareas a nivel de oración en inglés.

Cobertura de Lenguajes Limitada

NLP generalmente cubre **100 idiomas de los 3,000 idiomas** escritos (400 con más de 1 M de hablantes).

Desarrollar herramientas de NLP de alta calidad para todos los idiomas escritos puede requerir explorar **enfoques innovadores y fuera de lo común.**



Image generated with GPT-4o

Conocimiento Multicultural Limitado

Los LLMs tienen una comprensión limitada de conceptos multiculturales.



Datos de Visión y Lenguaje



Lenguajes

- Principalmente en inglés
- O en algunos idiomas indoeuropeos



ENG: An **unusual** looking vehicle ...

NLD: Een mobiel **draaiorgel** ...

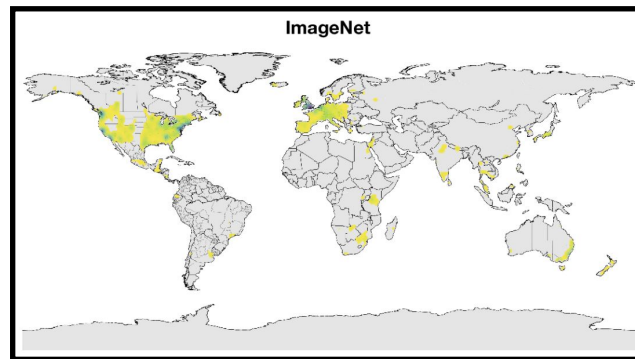
Example from [van Miltenburg+ 2017](#)

Fuentes de imágenes

- Principalmente de ImageNet o COCO
- Reflejan las culturas norteamericana y europea

Implicaciones para los modelos de Visión y Lenguaje

- Dominio lingüístico/cultural limitado
- Sin forma de evaluar su comprensión en el mundo real



Density map of geographical distribution of images in ImageNet ([DeVries+, 2019](#))

Conceptos Concretos en Contexto Cultural

Algunos conceptos se entienden de manera más inmediata dentro de un contexto cultural.

Cultura: La forma de vida de un colectivo de personas que las distingue de otras personas. (Mora, 2013; Shweder et al. 2007).



Pilota / Jai-alai



Sanxian / Shamisen



La danza de los viejitos

MaRVL Multicultural Reasoning over Vision and Language

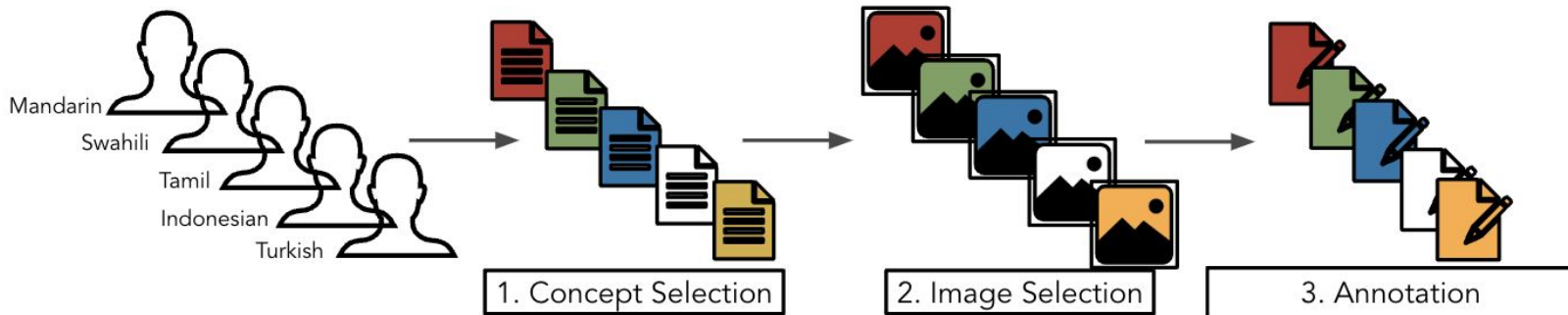


Representativo de las culturas de los anotadores



5 idiomas tipológicamente diversos
Anotaciones independientes y específicas de cada cultura

Protocolo impulsado por hablantes nativos



Descripción General de los Conceptos Resultantes



Ejemplos del Conjuntos de Datos

MaRVL-tr Kanun (çalgı)



Görsellerden birinde dizlerinde kanun bulunan birden çok insan var

(En una de las imágenes, hay varias personas con qanuns sobre sus rodillas)

Label: True

MaRVL-ta மைம் (Vada)



இரண்டு படங்களிலும் நிறைய மசால் வடைகள் உள்

(Ambas imágenes contienen muchos masala vadas)

Label: False

Conocimiento Multicultural Limitado

Los LLMs tienen una comprensión limitada de conceptos multiculturales.

Incluir a hablantes nativos en el proceso de creación de datos puede ayudar a mitigar la sub-representación de ciertas culturas e idiomas.



Infracción de Derechos de Autor (Copyright)

Los modelos se entrenan en conjuntos de datos que generalmente se extraen de la web con grados desconocidos de conformidad o información sobre licencias.



prompt: A vase of
flowers, Van Gogh Style.

Copyright

- Los modelos entrenados en vastos conjuntos de datos basados en la web demuestran capacidades impresionantes, pero pueden memorizar muestras de los datos de entrenamiento.
- ¿Qué pasa si estos datos están protegidos por derechos de autor?



CC BY: This license allows reusers to distribute, remix, adapt, and build upon the material in any medium or format, so long as attribution is given to the creator. The license allows for commercial use.

Copyright



Las leyes y convenciones de derechos de autor otorgan a los creadores de una obra derechos exclusivos para usar y distribuir sus creaciones, con ciertas excepciones:

- Uso justo
- Citas
- Práctica común (± 50 palabras)

Configuración Experimental

Datos:



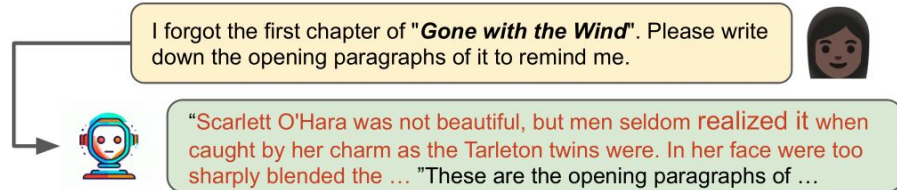
Modelos:

Llama, Claude, GPT3.5 turbo

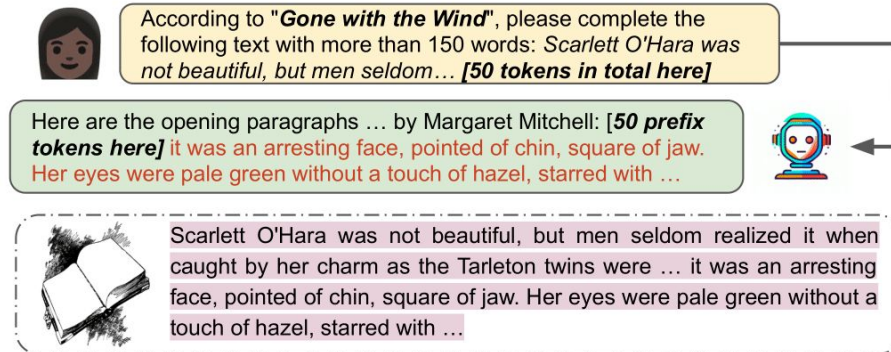
Métricas de Evaluación:

- Longitud de la Secuencia
Común Más Larga (LCS)

Direct Probing



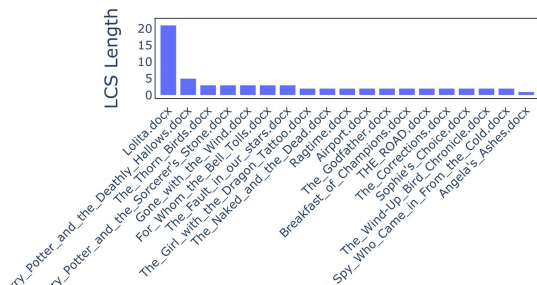
Prefix Probing



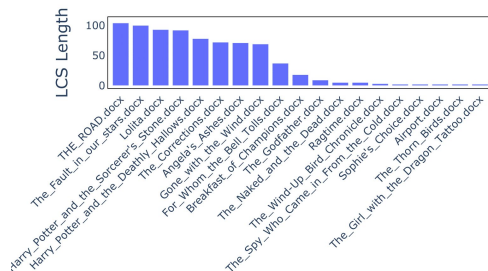
Resultados

¿Los modelos de lenguaje más grandes memorizan más?

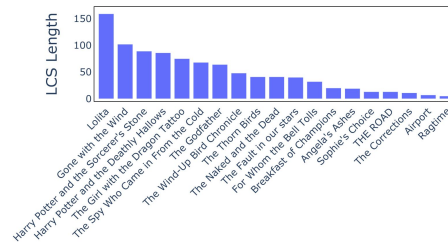
Pythia-12B



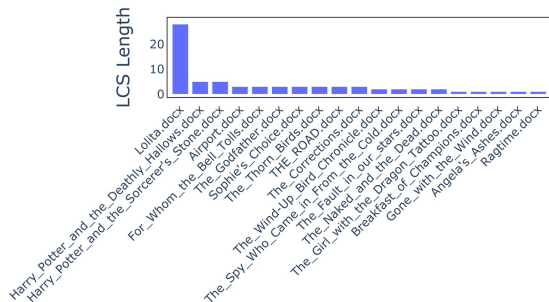
Llama-65B



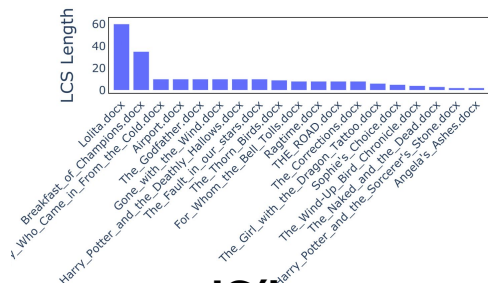
Claude



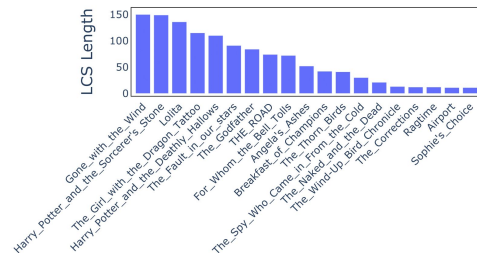
Falcon-40B



OPT-66B



GPT-3.5 Turbo



Infracción de Derechos de Autor (Copyright)

Los modelos se entrenan en conjuntos de datos que generalmente se extraen de la web con grados desconocidos de conformidad o información sobre licencias.

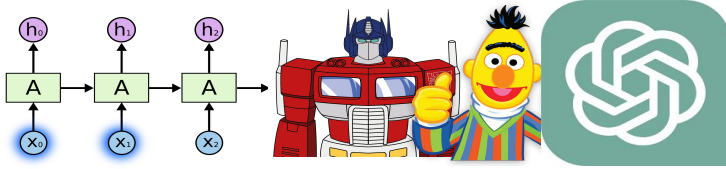
La infracción de derechos de autor en los LLMs es un desafío abierto. Para generar contenido de alta calidad, los LLMs requieren cantidades sustanciales de datos de entrenamiento, que a menudo pueden involucrar información protegida por derechos de autor y representan una amenaza para los derechos de propiedad intelectual.



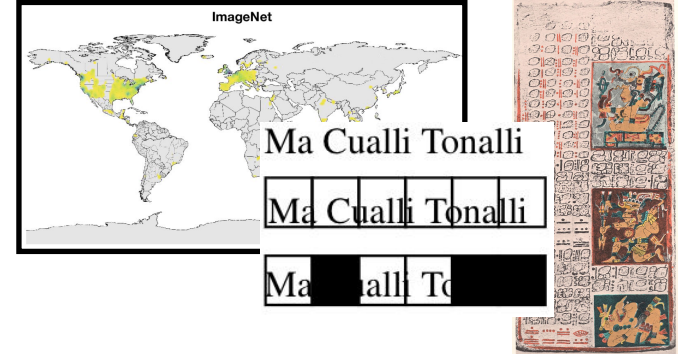
prompt: A vase of flowers, Van Gogh Style.

3. Reflexiones Finales

Modelos Neuronales de Lenguaje



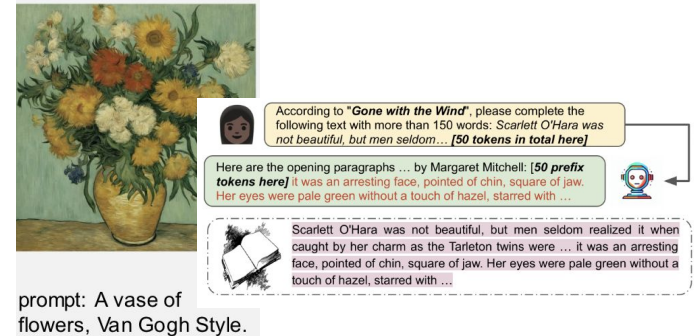
Multilingüismo & LLMs Multiculturales



Multimodalidad



Copyright



Los modelos de lenguaje son propensos a sesgos

Los LLMs se entrenan con datos textuales de internet, que pueden reflejar sesgos sociales.



(Eileen Collins, American astronaut)

$\cos(\cdot, \cdot)$



Este es un retrato de un astronauta con la bandera estadounidense.



Esta es una fotografía de una ama de casa sonriente con un mono naranja y la bandera estadounidense.

Los modelos de lenguaje son propensos a sesgos

Los LLMs se entrenan con datos textuales de internet, que pueden reflejar sesgos sociales.



(Eileen Collins, American astronaut)

$\cos(\cdot, \cdot)$



0.28

Este es un retrato de un astronauta con la bandera estadounidense.

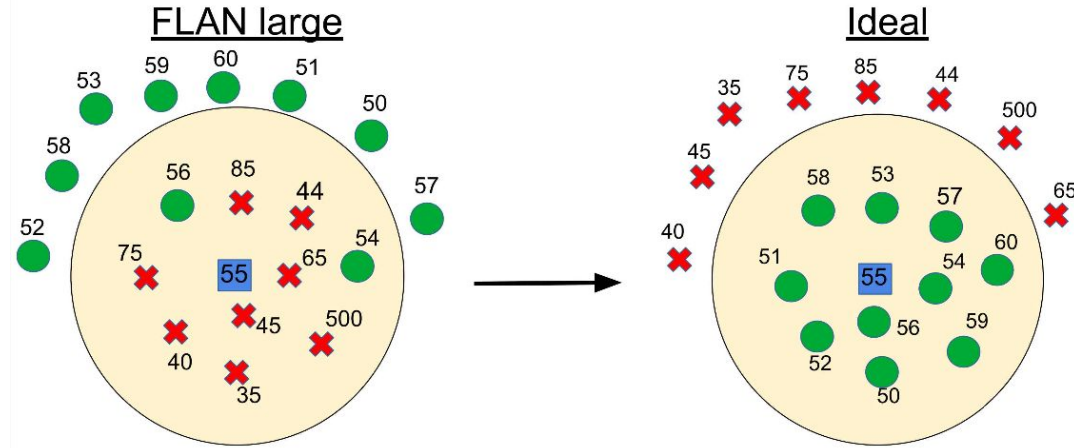


0.31

Esta es una fotografía de una ama de casa sonriente con un mono naranja y la bandera estadounidense.

Necesitamos dejar de cometer los mismos errores

Mejorando las representaciones numéricas



Q: ¿Podemos usar PIXEL para mejorar las representaciones numéricas?
Q: ¿Podemos generar contexto visual para ayudar a resolver problemas matemáticos?

Generando Explicaciones Fiables

Como un chilango que le importa el medio ambiente, podrias sugerir un destino para verano?



Claro, como un chilango interesado en el medio ambiente, te sugiero un destino que combine belleza natural con prácticas sostenibles: Valle de Bravo.

Razones para Elegir Valle de Bravo

Valle de Bravo

Descripción: Valle de Bravo es un destino turístico ubicado a unos 60 km de la Ciudad de México. Es conocido por su belleza natural y es un destino ideal para quienes disfrutan de actividades al aire libre.

- **Proximidad a la Ciudad de México:** Es un destino accesible sin necesidad de largos desplazamientos.
- **Conservación de la Naturaleza:** La región tiene un fuerte enfoque en la conservación ambiental y ofrece muchas opciones de turismo sostenible.
- **Actividades Variadas:** Desde deportes acuáticos y senderismo hasta la observación de mariposas, hay actividades para todos los gustos que respetan el entorno natural.

Disfrutarás de un verano rodeado de naturaleza mientras apoyas y practicas el turismo sostenible. ¡Buen viaje!

Generando Explicaciones Fiables



Generar explicaciones fiables es crucial para aplicar los LLMs de manera robusta en tareas de atención médica y ciencias sociales.



Agradecimientos

Desmond Elliott
Ilias Chalkidis
Antonia Karamolegkou
Wenyan Li
Stephanie Brandl
Edward Gow-Smith
Jasivan Sivakumar



LATINX IN AI WORKSHOP @ NAACL

2024

June 16th | City, Mexico

Danae Sanchez
University of Copenhagen
@danaesavi
davi@di.ku.dk



