¿Que nos falta?

jganzaba@itba.edu.ar

GPT

Generative Pre-Trained Transformer

Hoy:

- Transformers
- Hugging Face
- Entrenamiento y funcionamiento de modelos con Transformers

Próxima clase:

- Aplicaciones con LLMs
- Distintos LLMs
- Langchain y base de datos vectoriales (Q&A, summarización, etc)
- Functions
- RLHF

GPT

Generative Pre-Trained Transformer

GPT

Generative Pre-Trained **Transformer**

Transformers

Intro

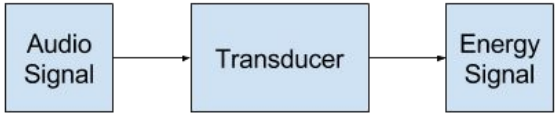
El transformer y sus características

"**Transformer** is the first **transduction** model relying entirely on **self-attention** to compute representations of its **input** and **output** without using sequencealigned **RNNs** or **convolution**"

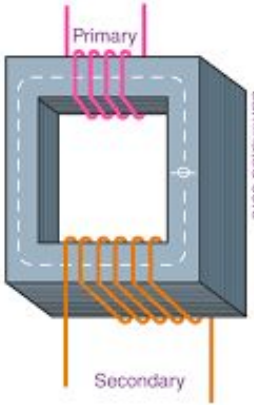
Tipo de arquitectura para modelos de deep learning

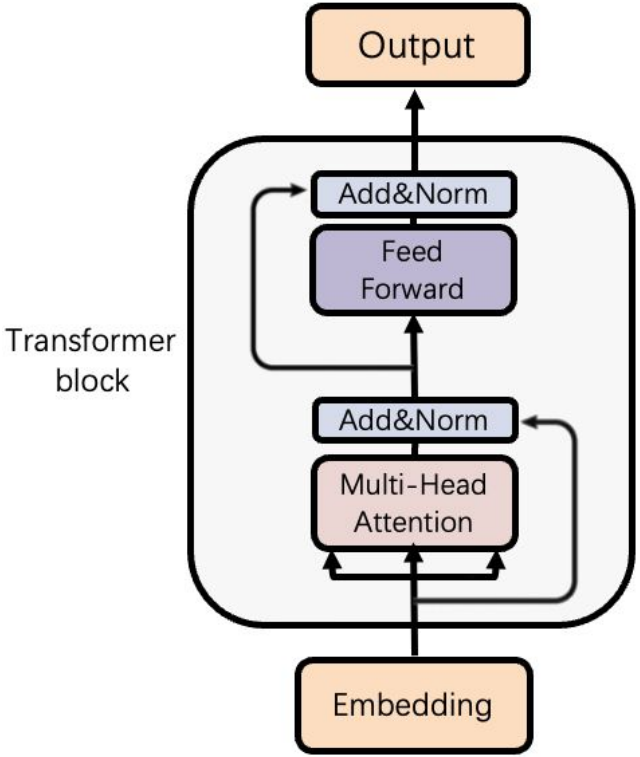
- Mecanismo de auto-atención (**self-attention**)
- Procesamiento en paralelo (GPU) vs LSTMs (Eficientes en entrenamiento)
- Transfer Learning / fine-tuning
- Embeddings **contextuales**
- Secuencias largas (Comparado con LSTM) - Tamaño de contexto (tokens)
- Alta performance en tareas de NLP y otras disciplinas
- Desarrollado por Google en el 2017

¿Por que transformer?

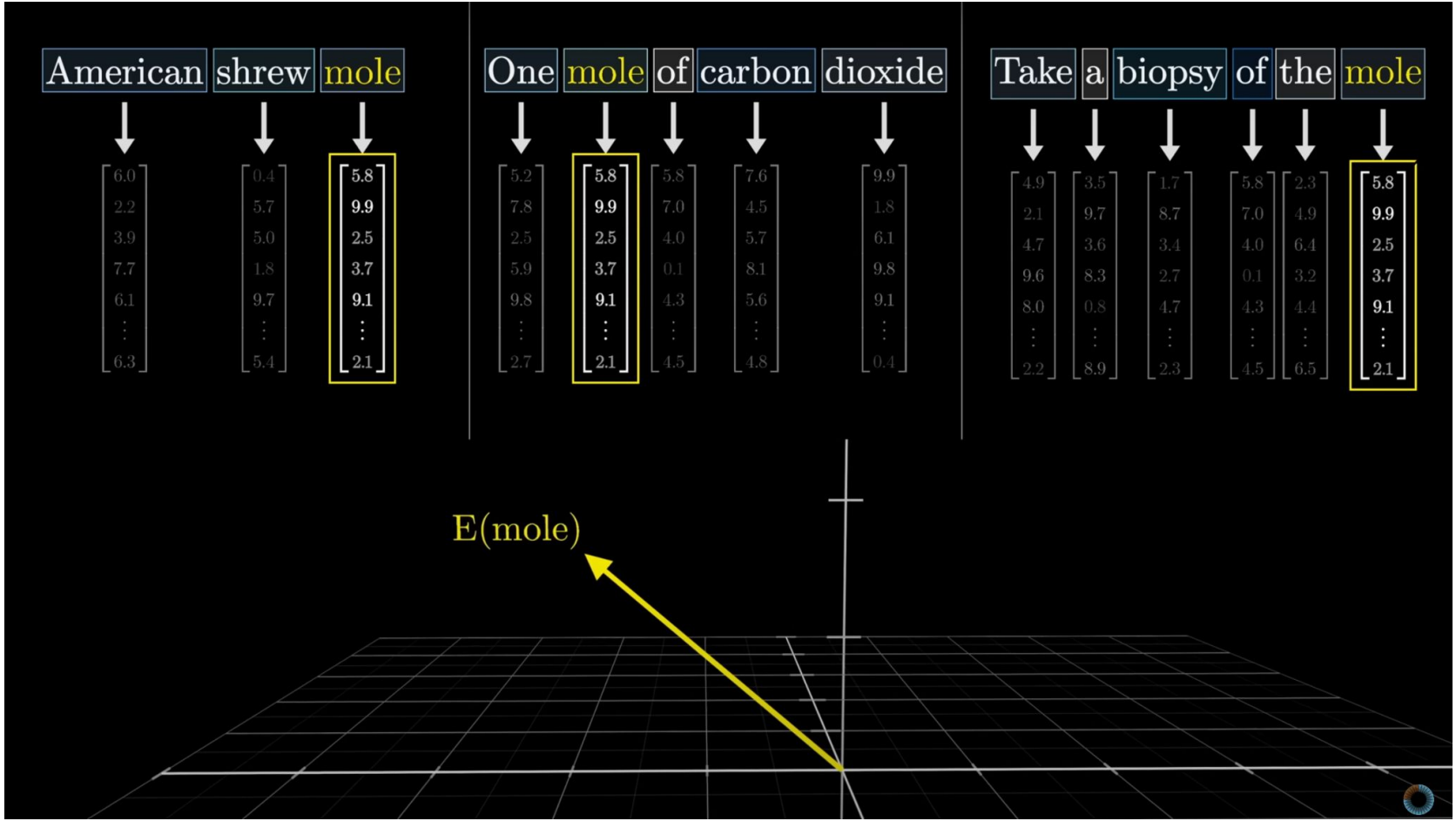


- Varias transformaciones
- Misma dimensión a la entrada que a la salida
- Marketing
- BERT (Plaza sésamo) - Google 2019

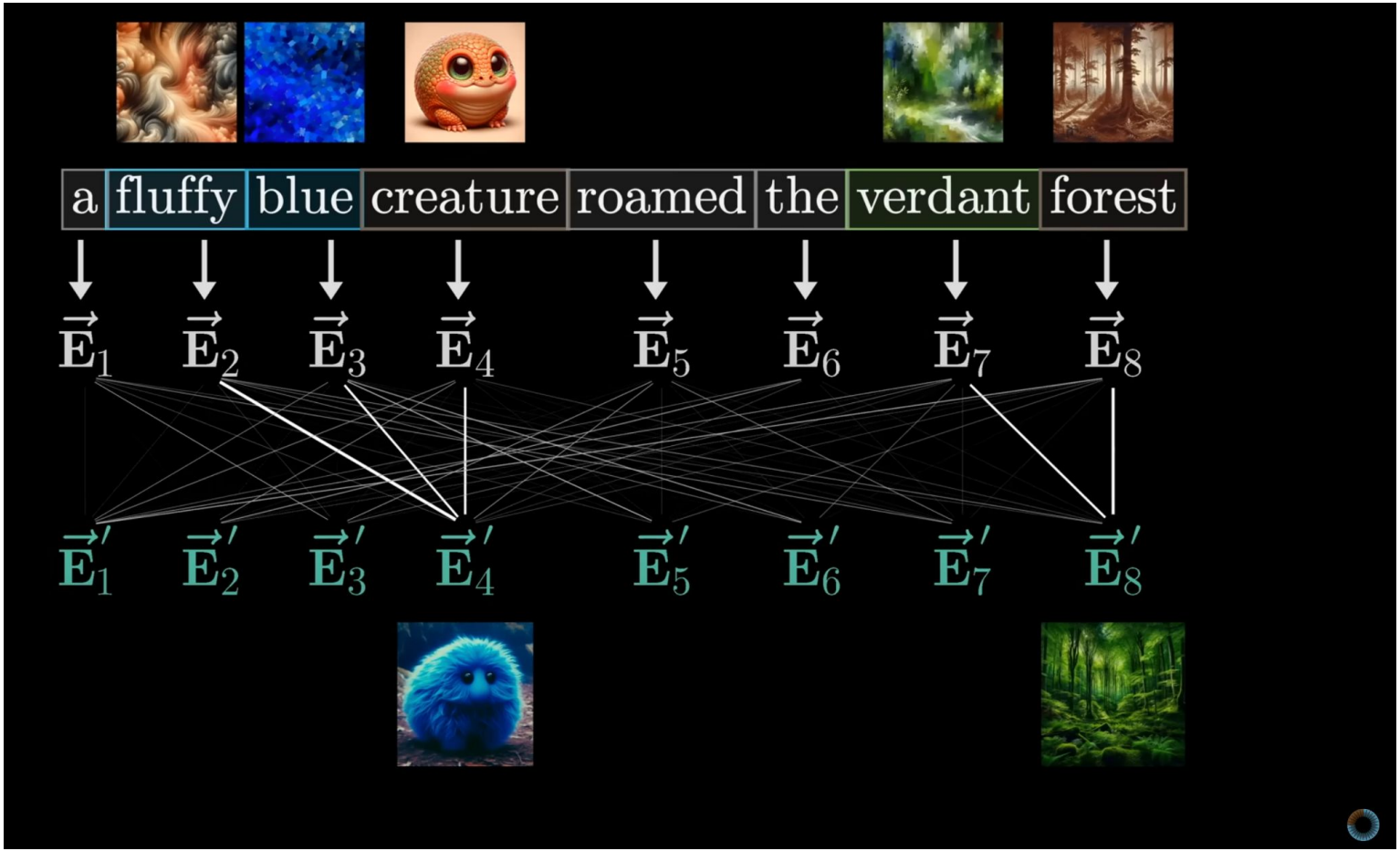




Embeddings **NO** contextuales (3blue1brown)



Embeddings contextuales (3blue1brown)



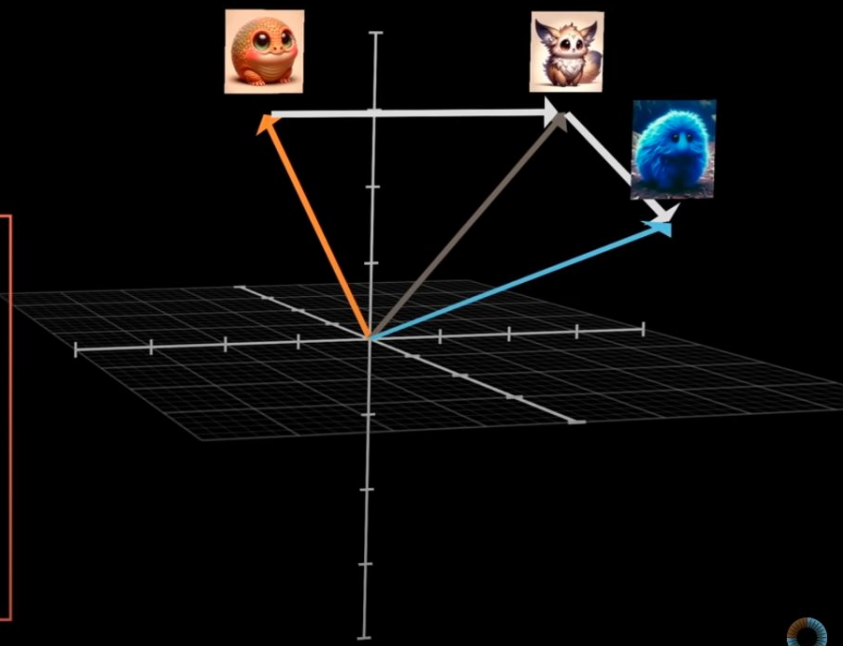
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{\boxed{QK^T}}{\sqrt{d_k}}\right)V$$

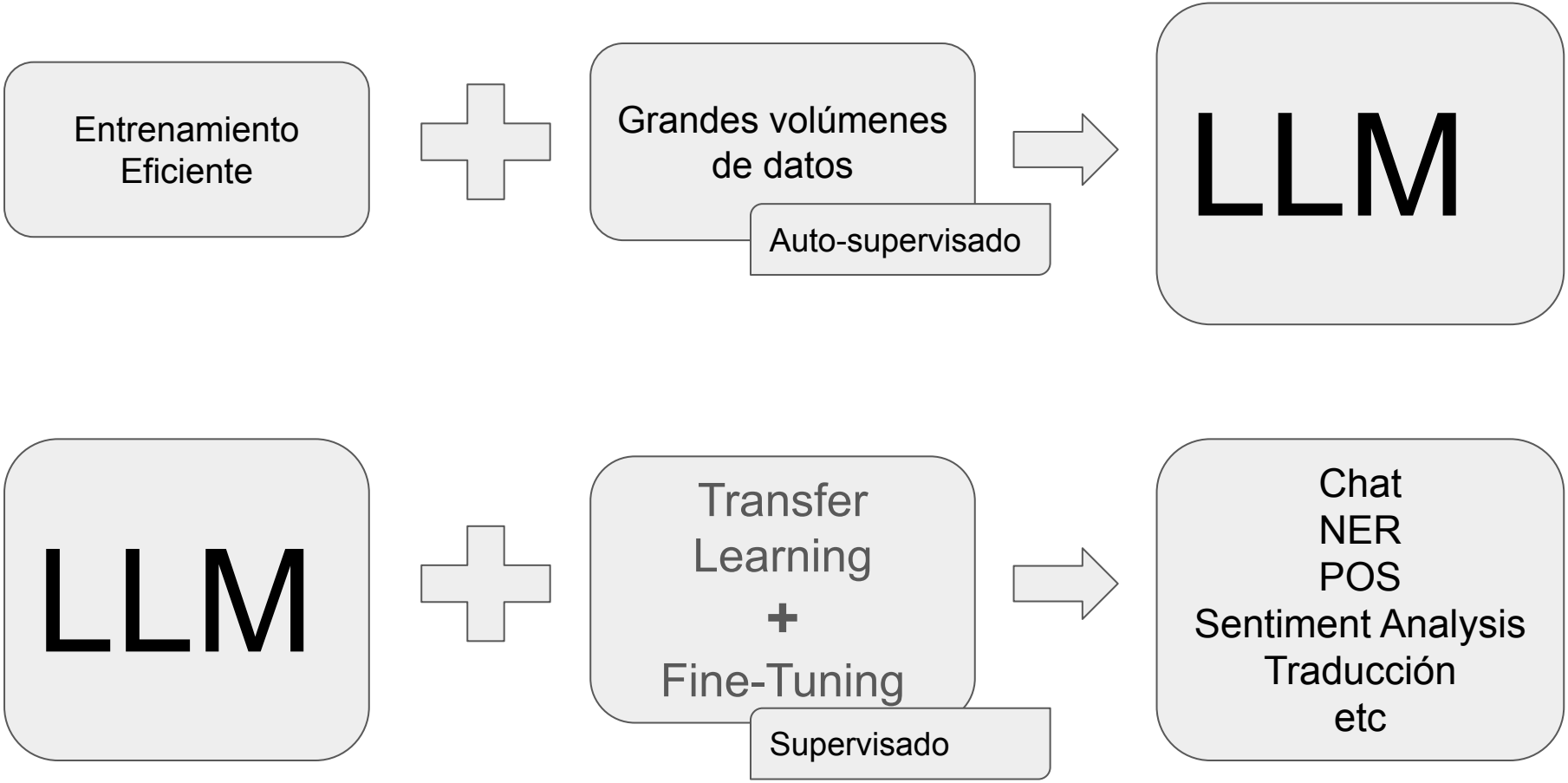
	Q_1	Q_2	Q_3	Q_4	Q_5	$\cdots$	Q_n
K_1	$Q_1 \cdot K_1$	$Q_2 \cdot K_1$	$Q_3 \cdot K_1$	$Q_4 \cdot K_1$	$Q_5 \cdot K_1$	$\cdots$	$Q_n \cdot K_1$
K_2	$Q_1 \cdot K_2$	$Q_2 \cdot K_2$	$Q_3 \cdot K_2$	$Q_4 \cdot K_2$	$Q_5 \cdot K_2$	$\cdots$	$Q_n \cdot K_2$
K_3	$Q_1 \cdot K_3$	$Q_2 \cdot K_3$	$Q_3 \cdot K_3$	$Q_4 \cdot K_3$	$Q_5 \cdot K_3$	$\cdots$	$Q_n \cdot K_3$
K_4	$Q_1 \cdot K_4$	$Q_2 \cdot K_4$	$Q_3 \cdot K_4$	$Q_4 \cdot K_4$	$Q_5 \cdot K_4$	$\cdots$	$Q_n \cdot K_4$
K_5	$Q_1 \cdot K_5$	$Q_2 \cdot K_5$	$Q_3 \cdot K_5$	$Q_4 \cdot K_5$	$Q_5 \cdot K_5$	$\cdots$	$Q_n \cdot K_5$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$

blue fluffy creature

$$\begin{bmatrix} +1.0 \\ +4.3 \\ +2.0 \\ +0.9 \\ -1.5 \\ +2.9 \\ \vdots \\ +7.8 \end{bmatrix} \quad \begin{bmatrix} +9.2 \\ -2.3 \\ +5.8 \\ +0.6 \\ +1.3 \\ +8.4 \\ \vdots \\ -8.2 \end{bmatrix} \quad \begin{bmatrix} -7.6 \\ +2.8 \\ -7.1 \\ +8.8 \\ +0.4 \\ -1.7 \\ \vdots \\ -4.7 \end{bmatrix}$$

$$\overbrace{\begin{bmatrix} -3.6 & -1.7 & -8.6 & +3.8 & +1.3 & -4.6 & \cdots & -8.0 \\ +1.5 & +8.5 & -3.6 & +3.3 & -7.3 & +4.3 & \cdots & -6.3 \\ +1.7 & -9.5 & +6.5 & -9.8 & +3.5 & -4.6 & \cdots & +9.2 \\ -5.0 & +1.5 & +1.8 & +1.4 & -5.5 & +9.0 & \cdots & +6.9 \\ +3.9 & -4.0 & +6.2 & -2.0 & +7.5 & +1.6 & \cdots & +3.8 \\ +4.5 & +0.0 & +9.0 & +2.9 & -1.5 & +2.1 & \cdots & -3.9 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ +1.5 & +3.0 & +3.0 & -1.4 & +7.9 & -2.6 & \cdots & +7.8 \end{bmatrix}}^{W_V} \begin{bmatrix} +1.0 \\ +4.3 \\ +2.0 \\ +0.9 \\ -1.5 \\ +2.9 \\ \vdots \\ +7.8 \end{bmatrix} = \begin{bmatrix} -103.0 \\ +13.1 \\ +12.6 \\ +95.7 \\ +11.2 \\ +14.4 \\ \vdots \\ +62.1 \end{bmatrix}$$




Entrenamiento Eficiente
Grandes volúmenes de datos
Auto-supervisado
LLM
LLM
Transfer Learning
+
Fine-Tuning
Supervisado
Chat
NER
POS
Sentiment Analysis
Traducción
etc

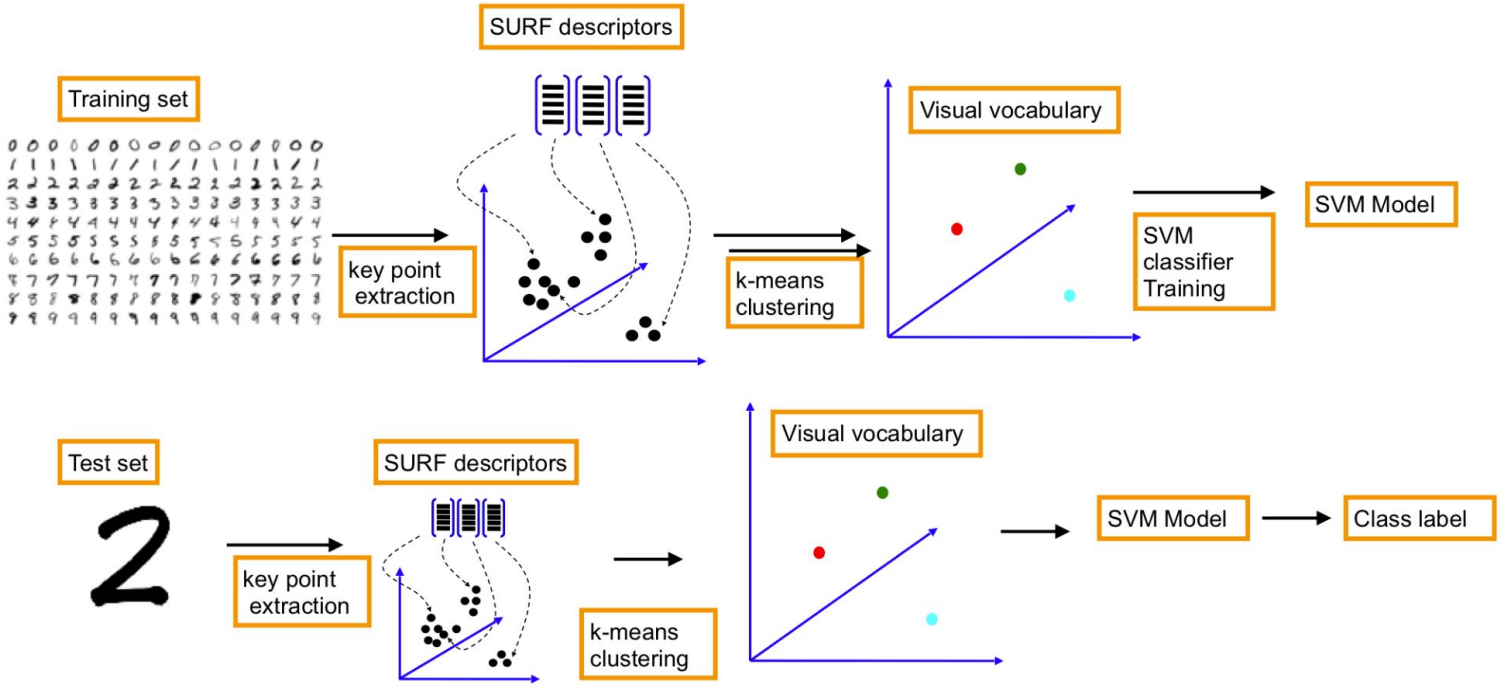
Reseña Histórica Deep Learning

Andrej Karpathy Video

Antes del 2012

- Extracción de features manuales
- Múltiples extractores
- SVM para combinarlos
- Algoritmo diferente para cada disciplina:
  - NLP
  - CV
  - RL

Antes del 2012



2012: Competencia Imagenet

Mucha data + Red Neuronal

→ Performance muy superior a estrategias anteriores

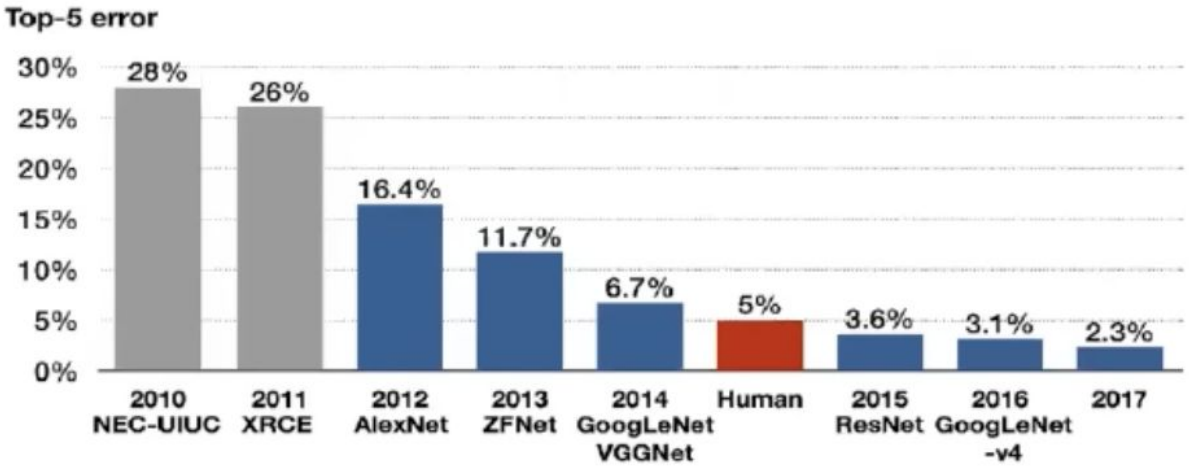
- **Data Anotada**
- Red Neuronal convolucional
- Aplicar esta arquitectura a otras disciplinas:
  - NLP
  - RL
- Universidad de Toronto (Hinton)

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

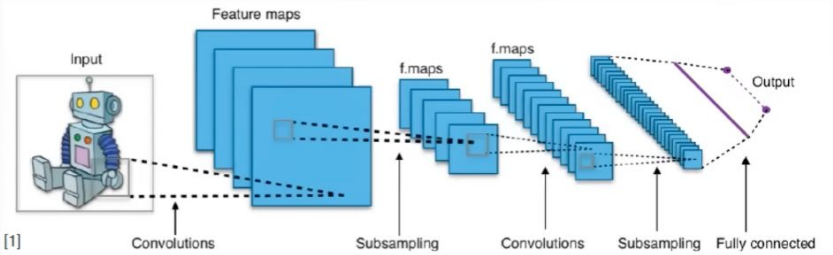
Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca



Neural nets started to take over AI. They all looked different…

Computer Vision

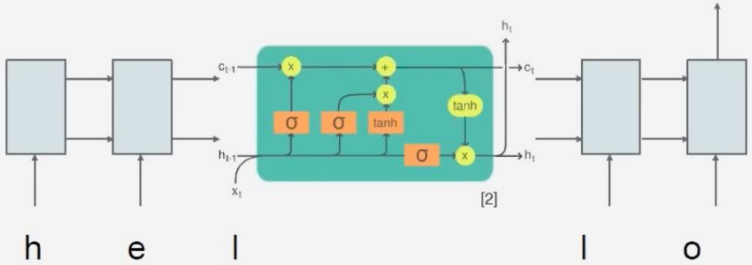
Convolutional NNs (+ResNets)



[1]

Natural Lang.

Recurrent NNs (+LSTMs)

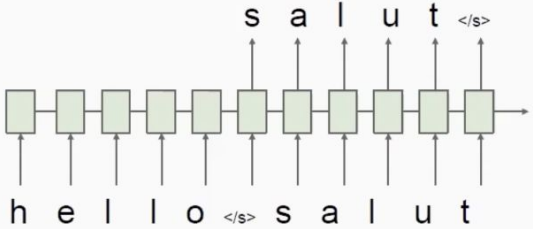


Speech

Deep Belief Nets (+non-DL)

Translation

Seq2Seq



RL

BC/GAIL

Algorithm 1 Generative adversarial imitation learning

1: **Input:** Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
2: **for** $i = 0, 1, 2, \ldots$ **do**
3: Sample trajectories $\tau_i \sim \pi_{\theta_i}$
4: Update the discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \quad (17)$$

5: Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)] - \lambda \nabla_\theta H(\pi_\theta), \quad \text{where } Q(\bar{s},\bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \mid s_0 = \bar{s}, a_0 = \bar{a}] \quad (18)$$

6: **end for**

Stanford

[1] CNN image CC-BY-SA by Aphex34 for Wikipedia https://commons.wikimedia.org/wiki/File:Typical_cnn.png
[2] RNN image CC-BY-SA by GChe for Wikipedia https://commons.wikimedia.org/wiki/File:The_LSTM_Cell.svg

slide from Lucas Beyer ty

2017 paper:
All you need is Attention

Foco en traducción de texto

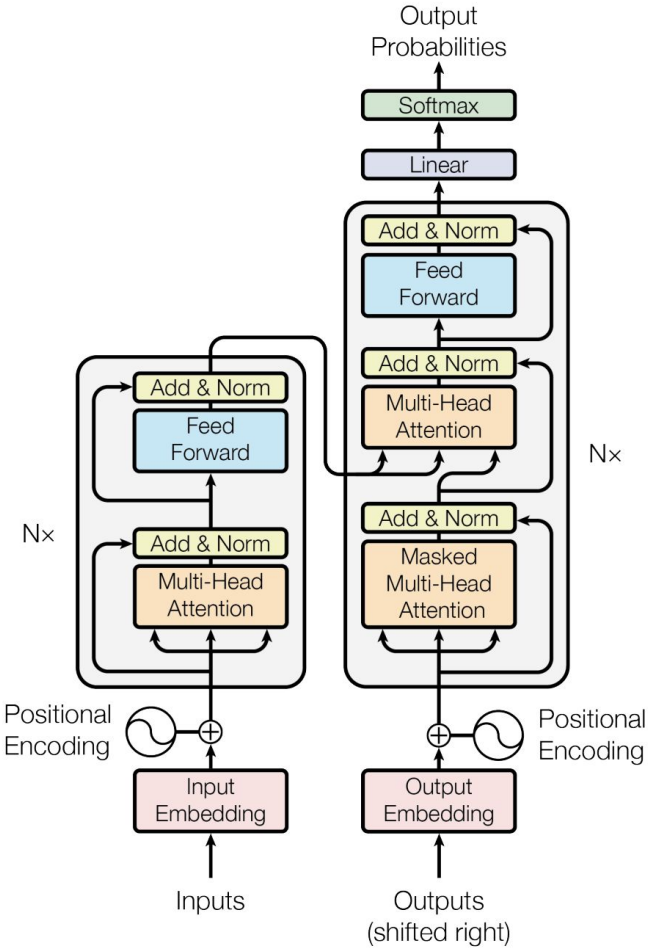
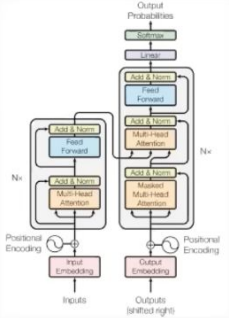
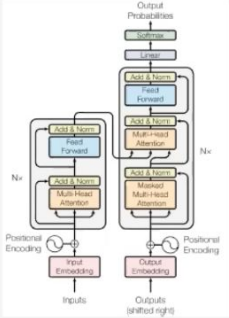


Figure 1: The Transformer - model architecture.

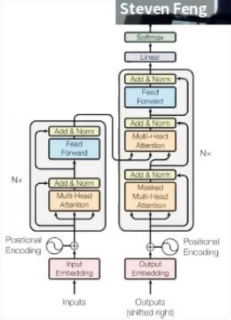
Computer Vision



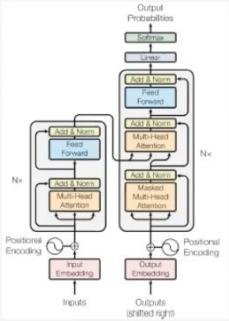
Natural Lang. Proc.



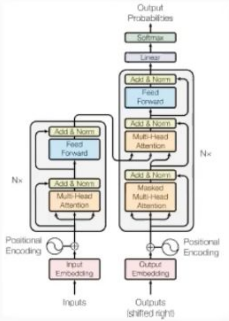
Reinf. L



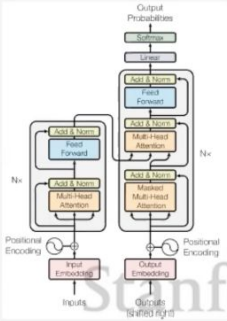
Speech



Translation



Graphs/Science



Transformer image source: "Attention Is All You Need" paper

Stanford

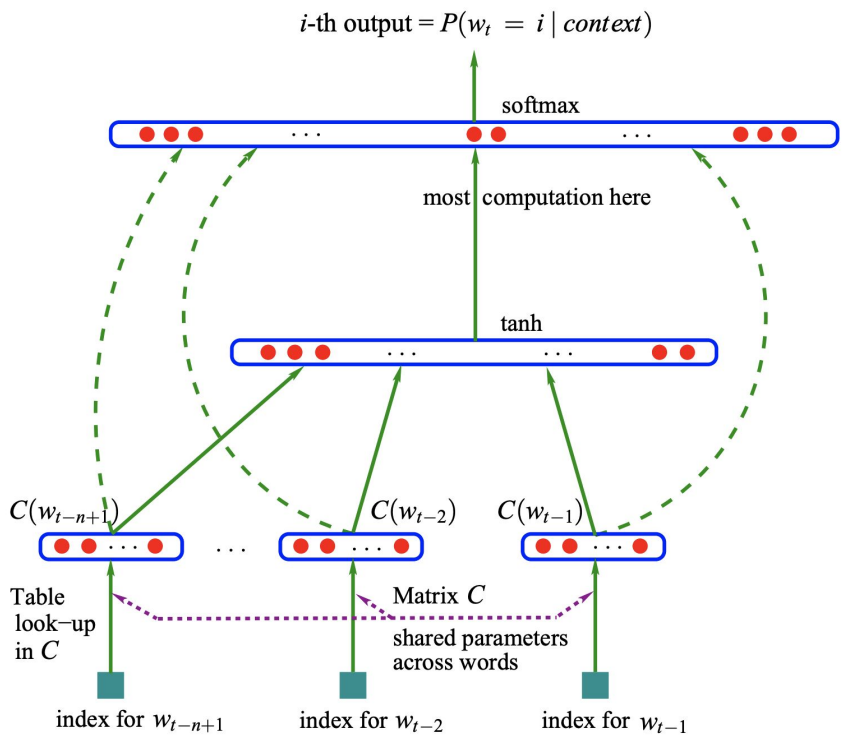
slide from Lucas Beyer ty

Resúmen

- Auto-supervisado (Billones de params → Entrenamiento restringido a grandes empresas por costos)
- Transfer Learning → Tarea específica
- Transformers muy eficientes GPU (Paralelización)

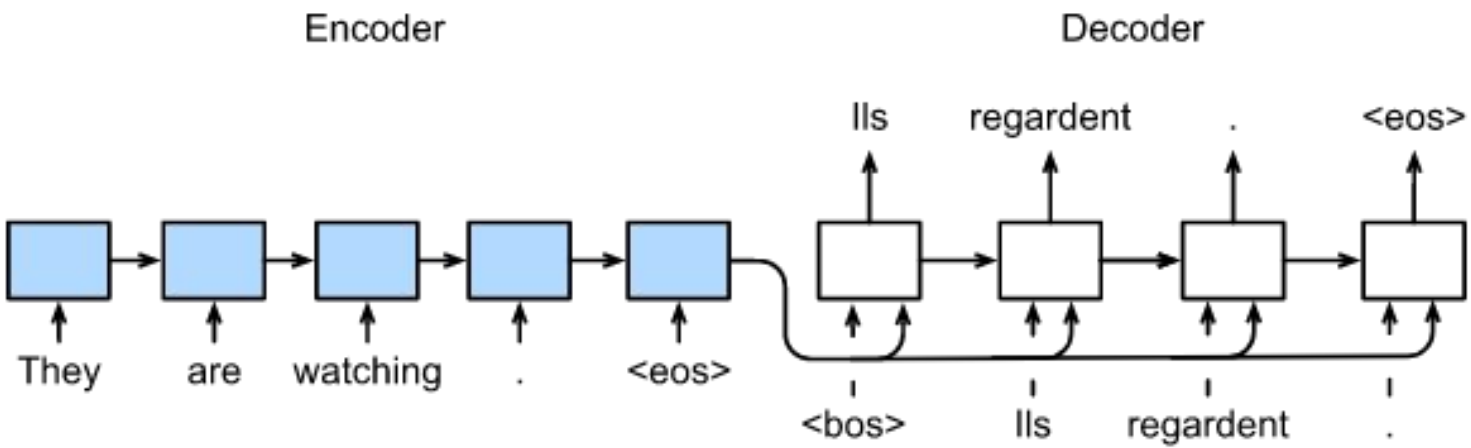
Reseña Histórica Transformers

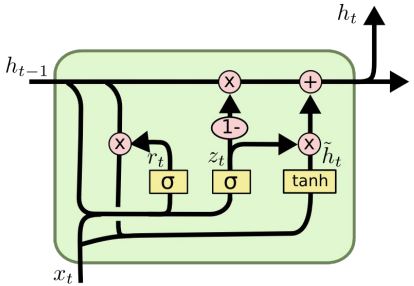
2003: A Neural probabilistic Language model



MLP para predecir la próxima palabra dado el contexto

2014: Sequence to Sequence Learning with Neural Nets





$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

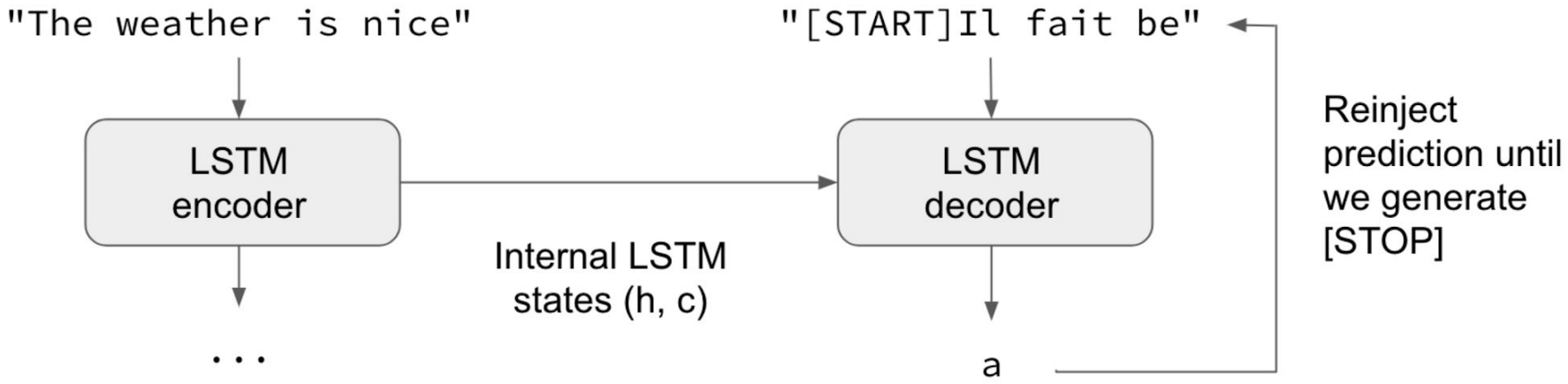
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

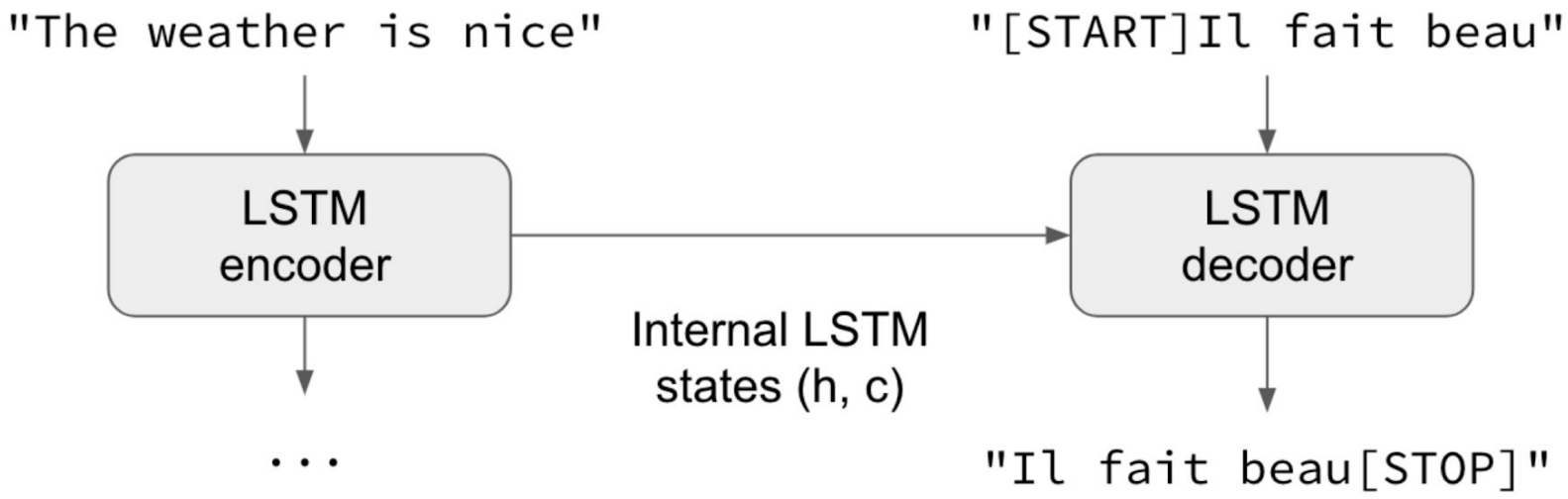
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Problema:
Cuello de botella del vector de contexto

Inferencia


"The weather is nice"
"[START]Il fait be"
LSTM encoder
LSTM decoder
Internal LSTM states (h, c)
...
a
Reinject prediction until we generate [STOP]

Entrenamiento



Encoder → Vector (Representa contexto)

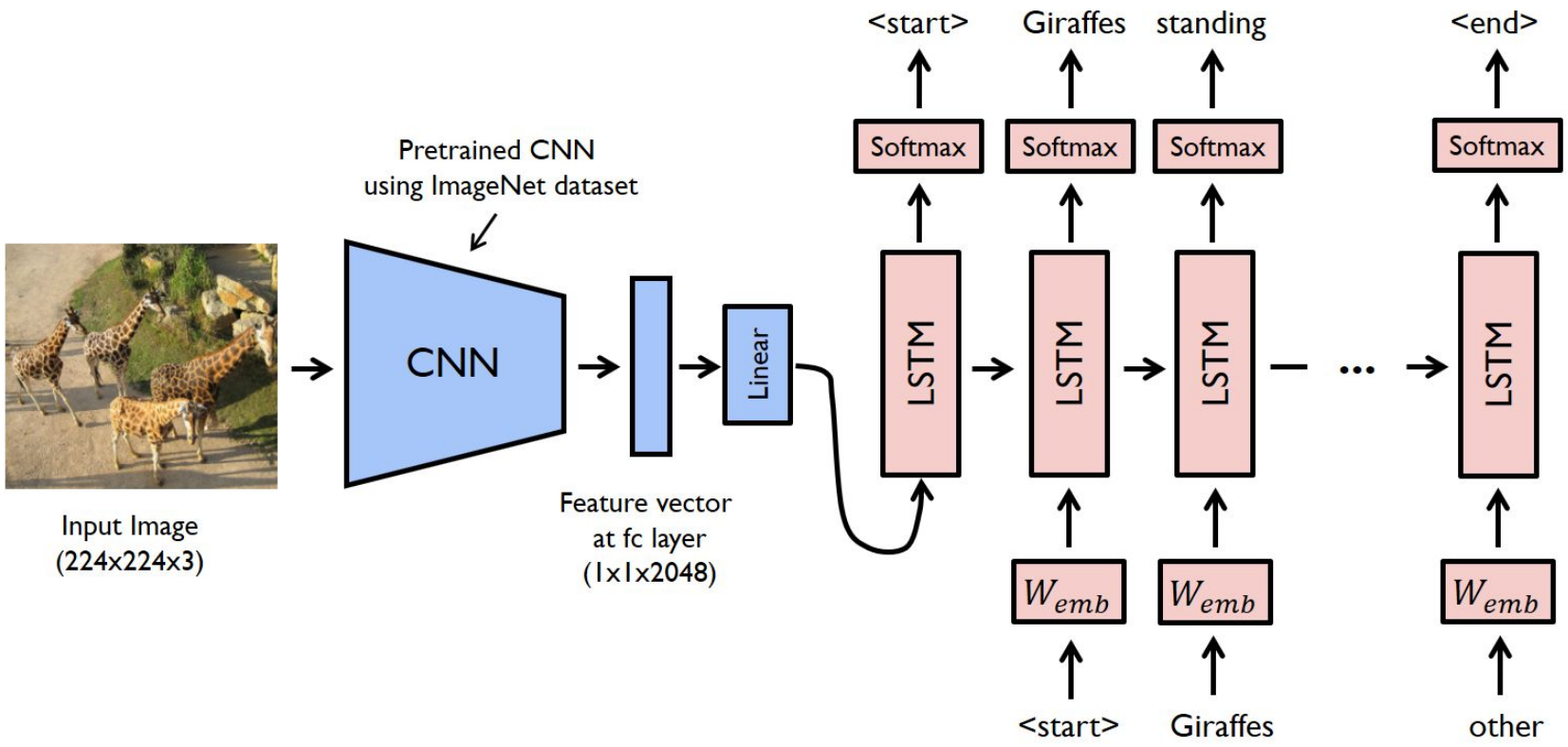
Decoder → Modelo de lenguaje

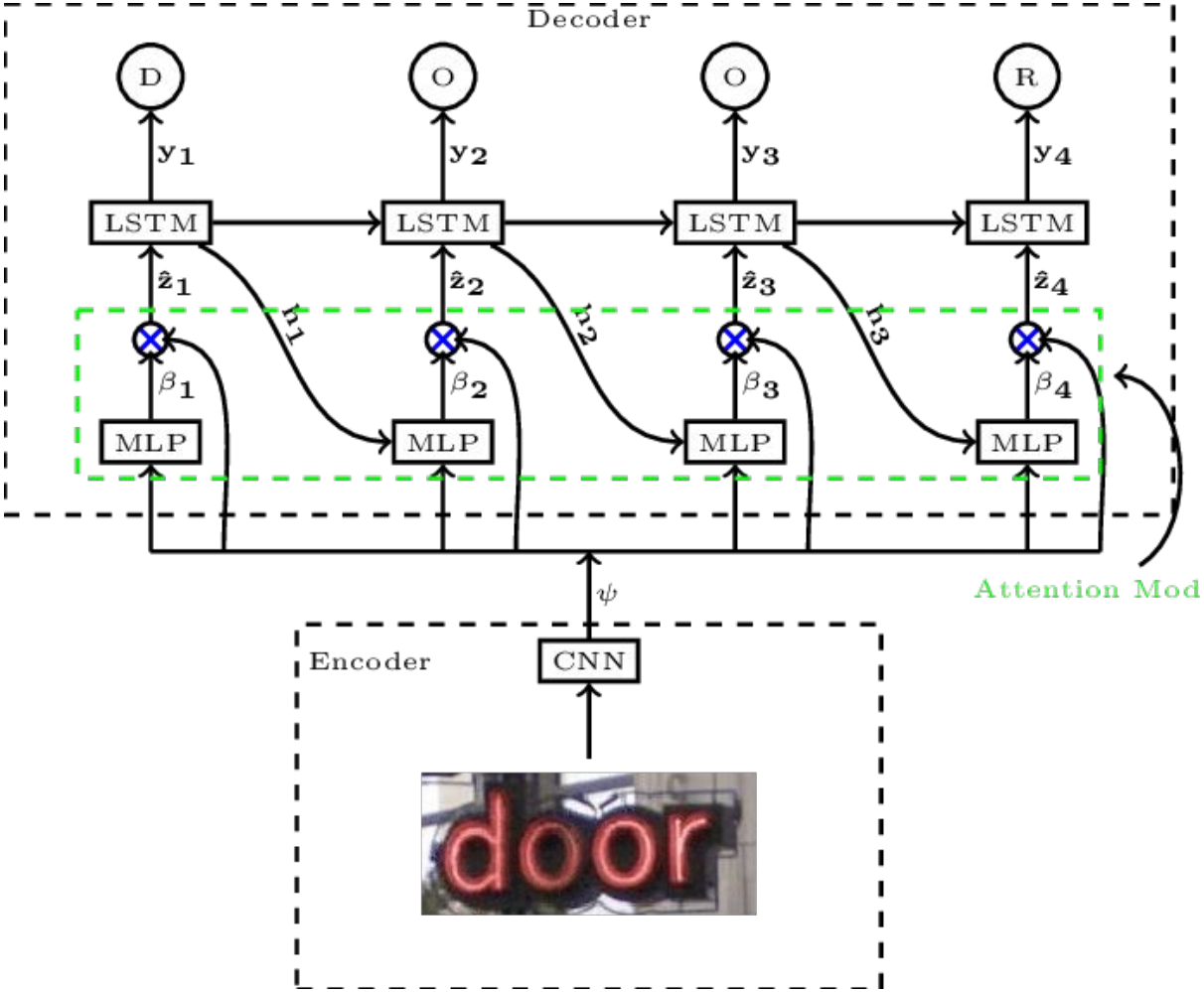
Traducción

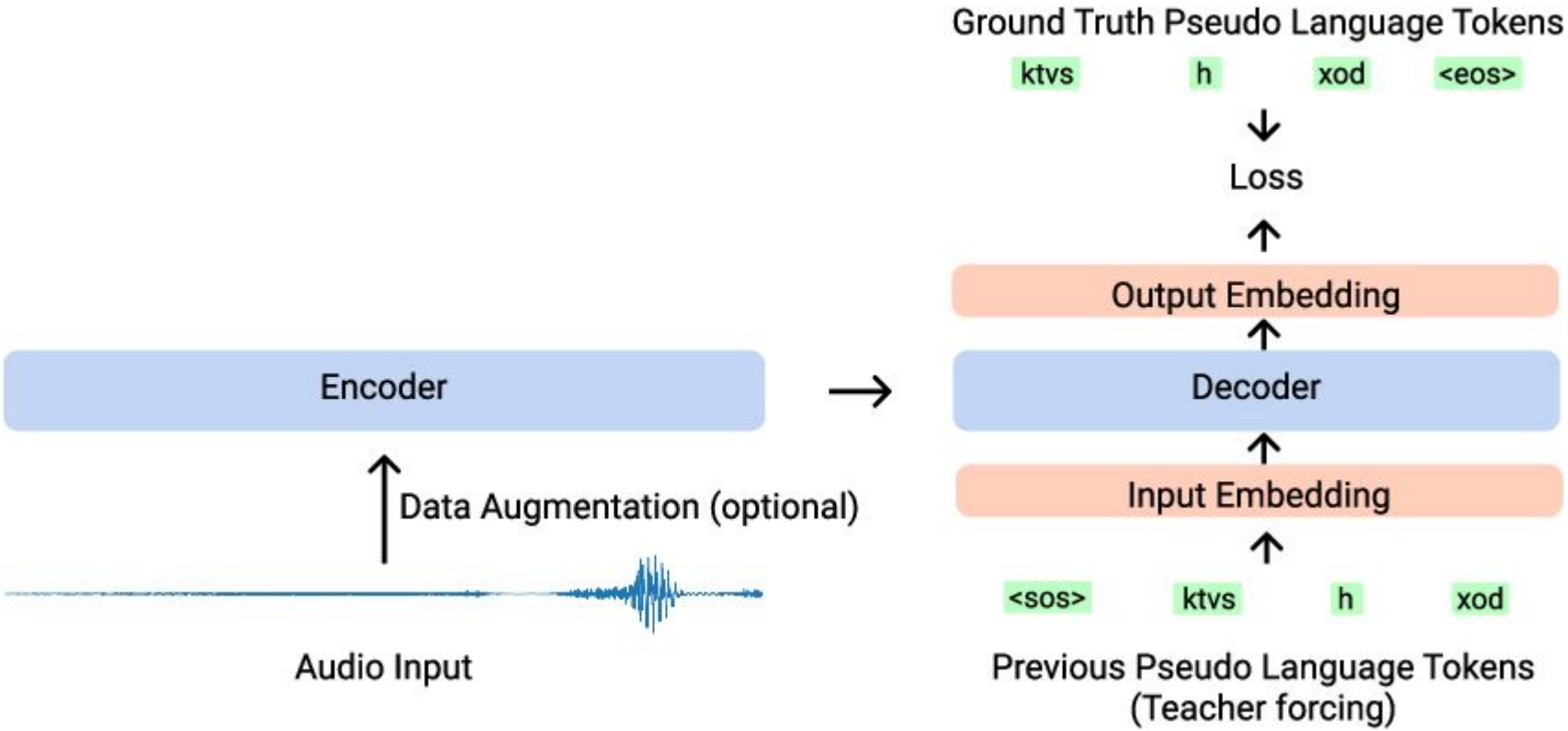
Image Captioning

Speech to text

OCR


Pretrained CNN
using ImageNet dataset
CNN
Input Image
(224x224x3)
Feature vector
at fc layer
(1x1x2048)
Linear
LSTM
LSTM
LSTM
LSTM
Softmax
Softmax
Softmax
Softmax
<start>
Giraffes
standing
<end>
W_{emb}
W_{emb}
W_{emb}
<start>
Giraffes
other


Decoder
D
O
O
R
y_1
y_2
y_3
y_4
LSTM
LSTM
LSTM
LSTM
$\hat{z}_1$
$\hat{z}_2$
$\hat{z}_3$
$\hat{z}_4$
h_1
h_2
h_3
β_1
β_2
β_3
β_4
MLP
MLP
MLP
MLP
ψ
Attention Mod
Encoder
CNN
door


Ground Truth Pseudo Language Tokens
ktvs
h
xod
<eos>
Loss
Output Embedding
Decoder
Input Embedding
<sos>
ktvs
h
xod
Previous Pseudo Language Tokens
(Teacher forcing)
Encoder
Data Augmentation (optional)
Audio Input

2014: NMT by Jointly learning to align and translate

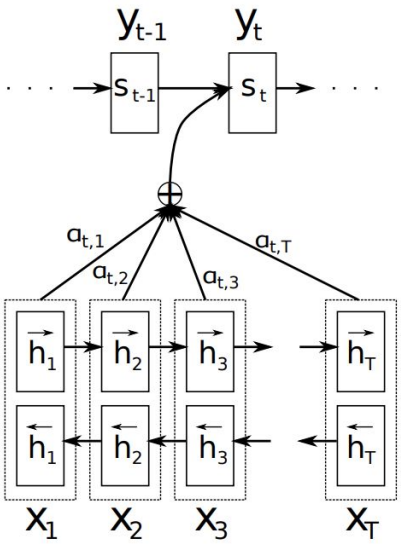


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

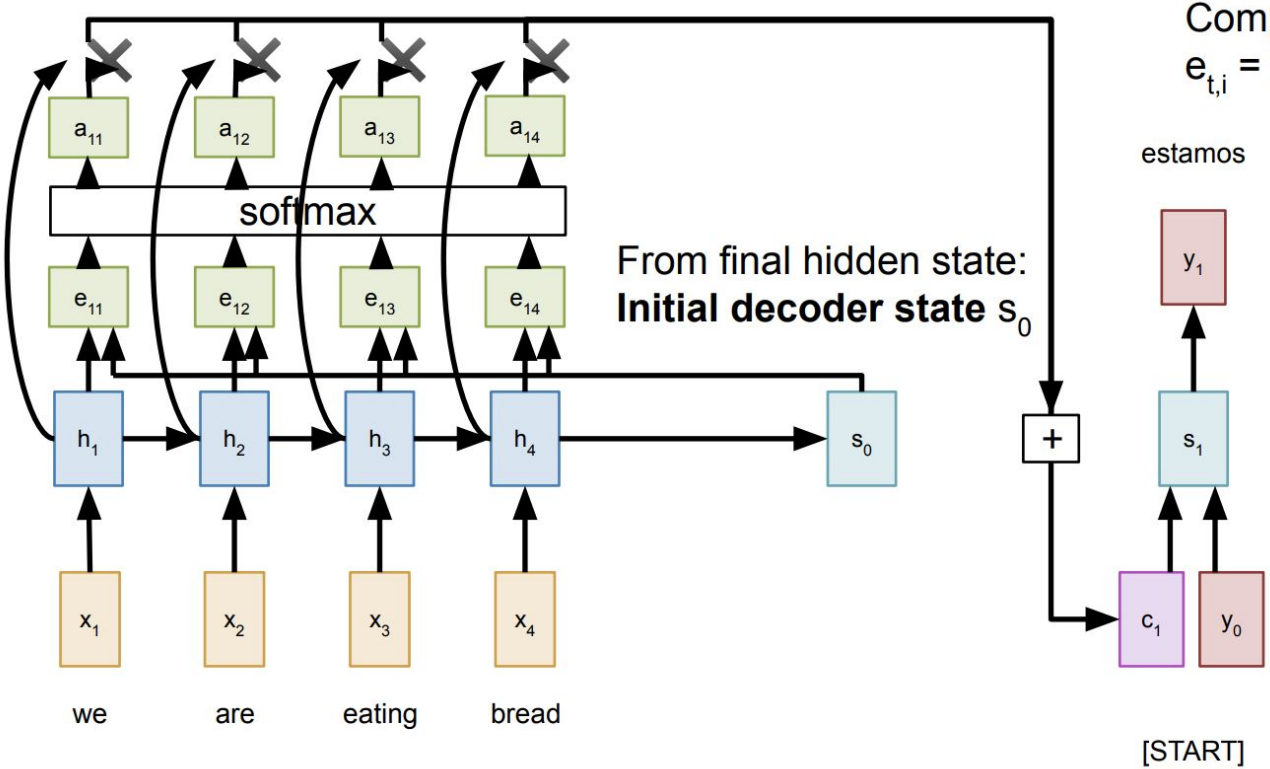
The weight α_{ij} of each annotation h_j is computed by

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

where

$$e_{ij} = a(s_{i-1}, h_j)$$

Explicabilidad

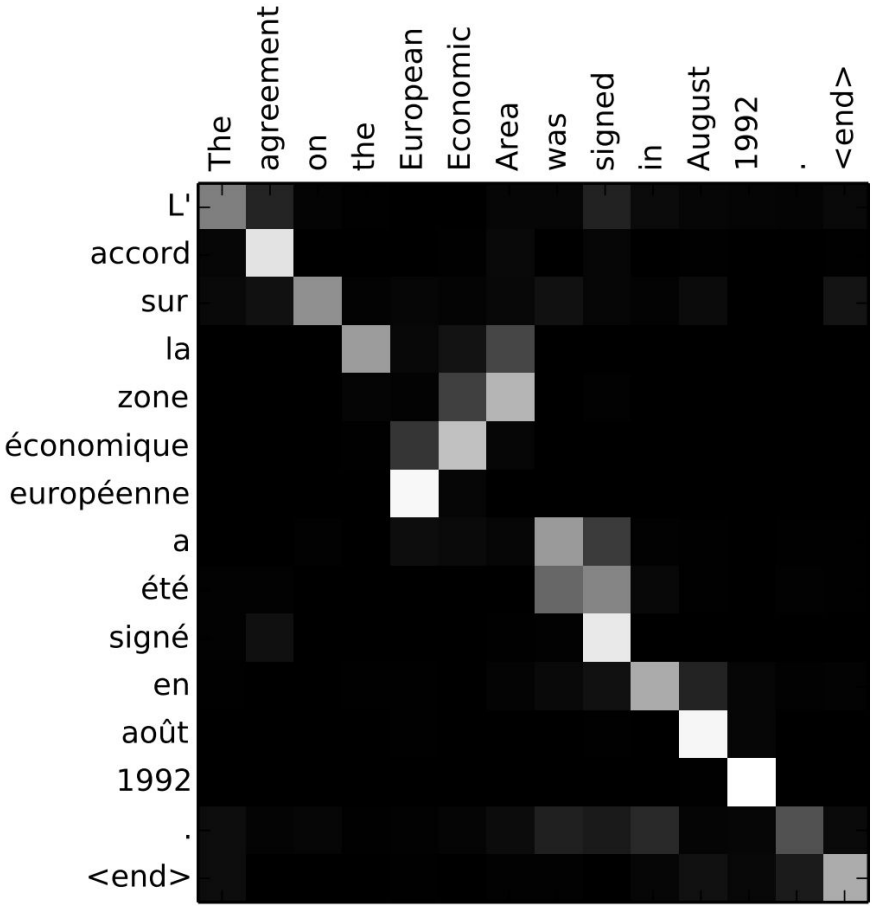


From final hidden state:
Initial decoder state s_0

Compute (scalar) **alignment scores**
$e_{t,i} = f_{att}(s_{t-1}, h_i)$ (f_{att} is an MLP)

Normalize alignment scores to get **attention weights**
$0 < a_{t,i} < 1$ $\sum_i a_{t,i} = 1$

Compute context vector as linear combination of hidden states
$c_t = \sum_i a_{t,i} h_i$


The
agreement
on
the
European
Economic
Area
was
signed
in
August
1992
.
<end>
L'
accord
sur
la
zone
économique
européenne
a
été
signé
en
août
1992
.
<end>

El problema del cuello de botella se potencia en imágenes → Attention in images

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

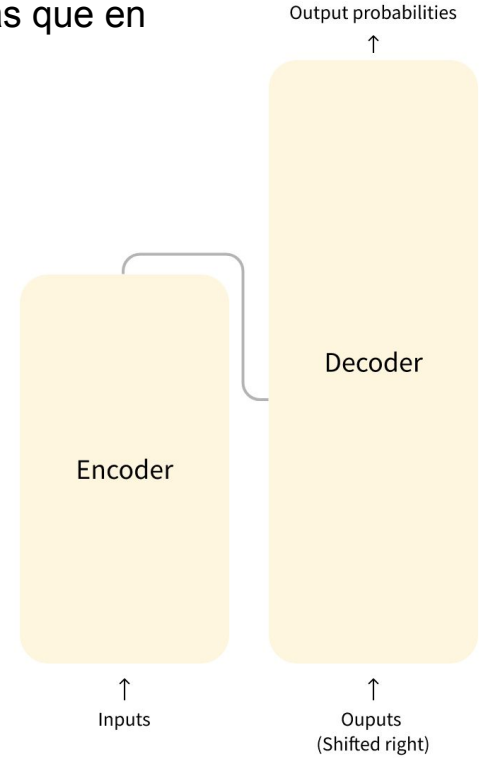
Transformers

Only Attention

2017 paper: All you need is Attention

Combinación de muchas técnicas que en conjunto funcionan muy bien:

- Attention
- Sin RNNs (LTSM)
- Residual Networks
- Positional autoencoding
- Multi-head
- MLP
- Normalización
- Encoder - decoder



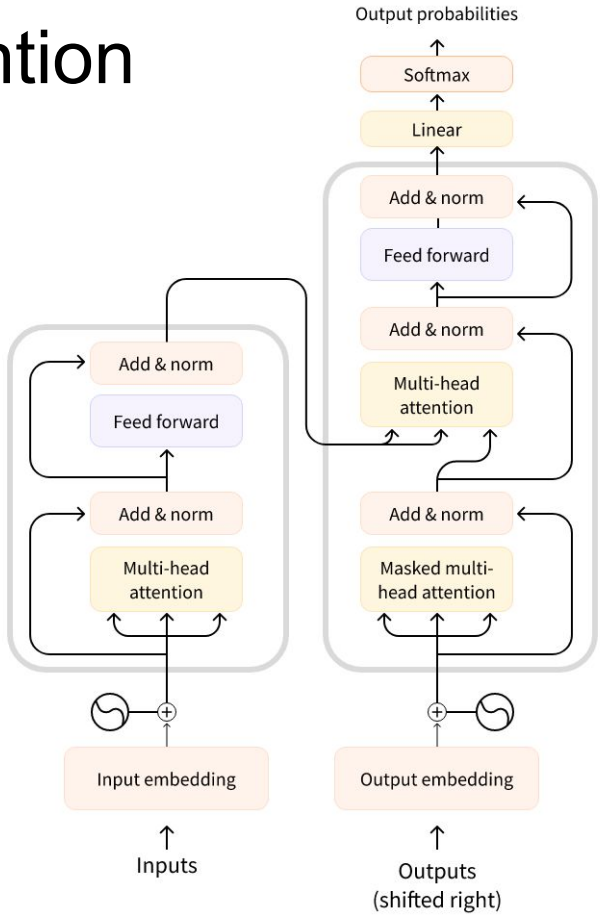


Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

FLOPS = Floating Point Operations (Se estima)

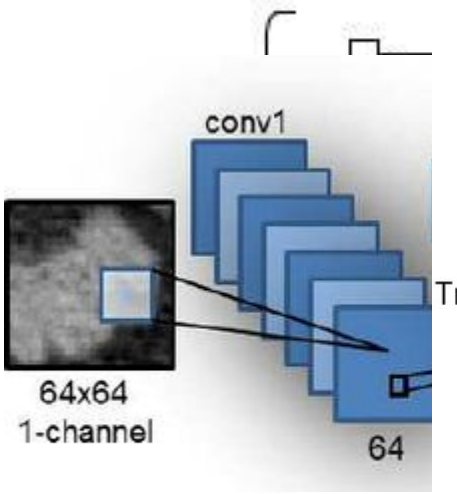
"We estimate the number of **floating point operations** used to train a model by multiplying the t**raining time**, the **number of GPUs** used, and an estimate of the sustained single-precision floating-point capacity of each GPU"

"Training took 3.5 days on 8 P100 GPUs"

"beam size of 4 and length penalty α = 0.6"

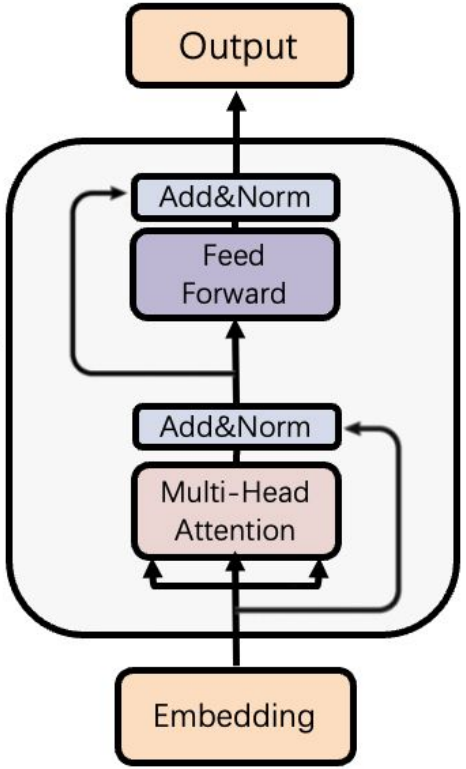
Cómo crecer en complejidad

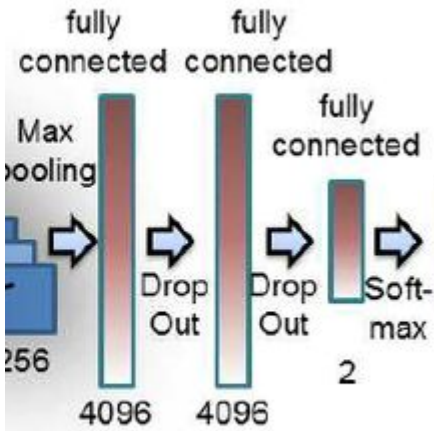
Horizontal vs Vertical



Input Layer

Transformer block





Output Layer

Explicabilidad

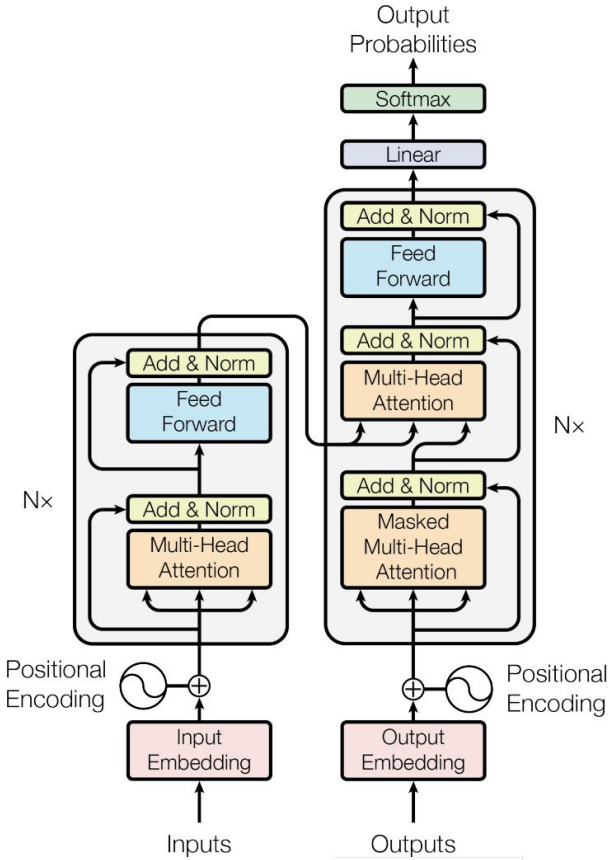
“Las razones por las que el modelo dio cierto resultado se puede explicar/entender”

- Cada bloque de **transfomer** tiene su mecanismo de **attention**
- Cada mecanismo de attention se aplica sobre su entrada, que es la salida del bloque anterior + Multi-head

Esto complica la explicabilidad y la intuición sobre qué está haciendo realmente el transformer

BERT

Encoder

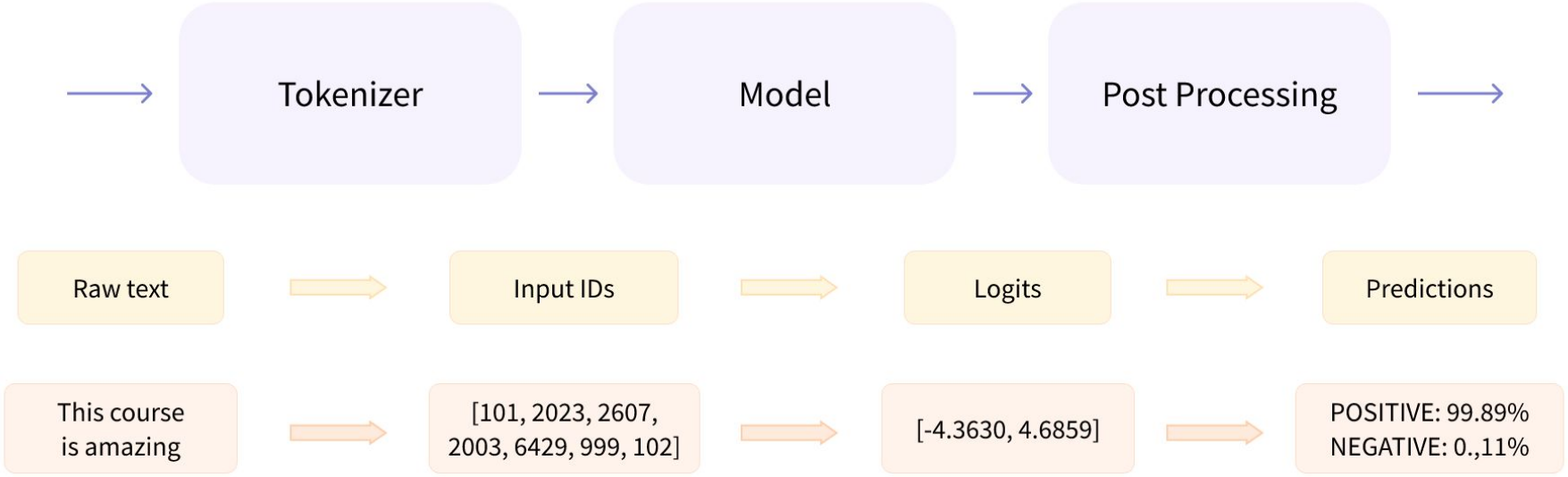


GPT

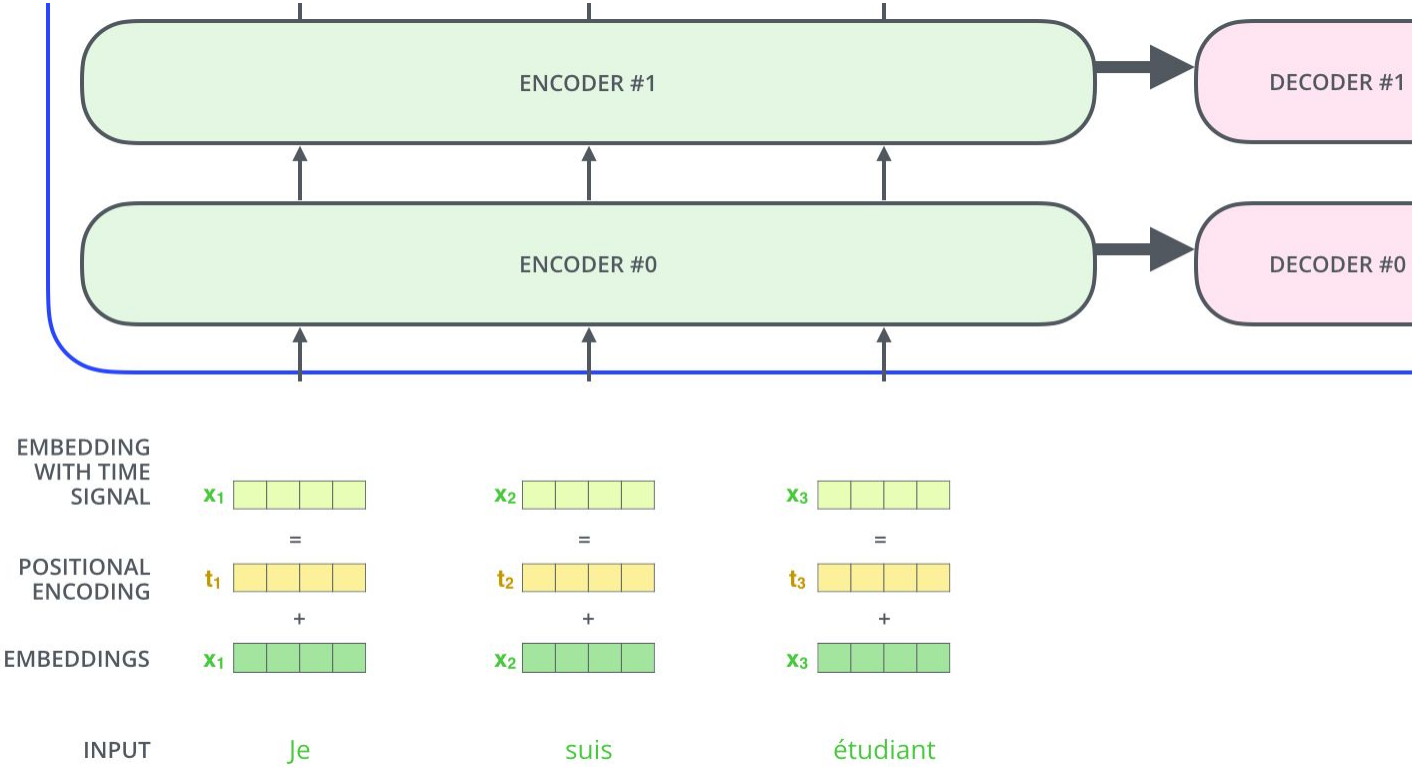
Decoder

BERT: Bidirectional Encoder Representations from Transformers

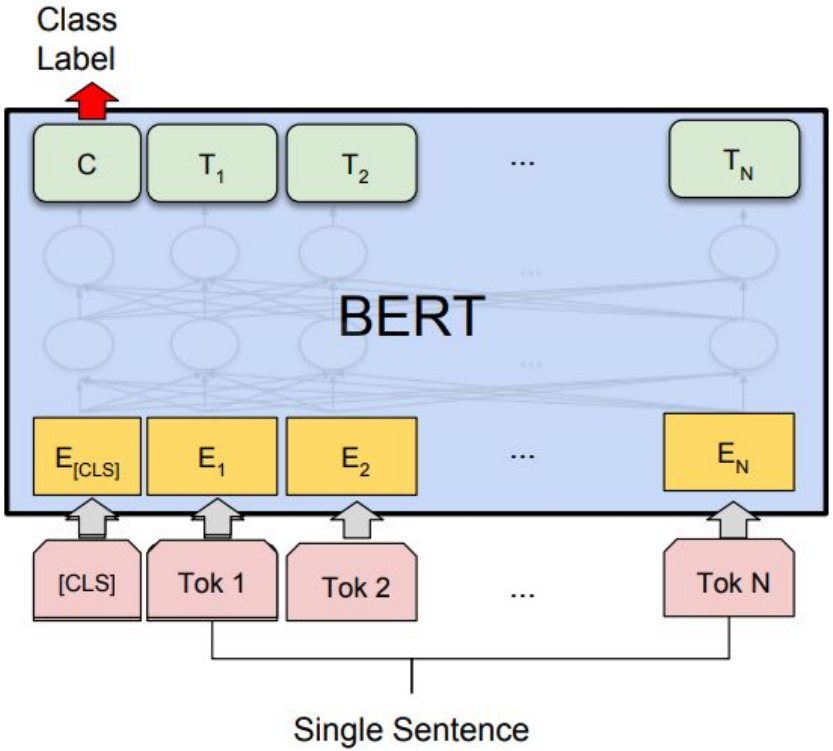
Pipeline


Tokenizer
Model
Post Processing
Raw text
Input IDs
Logits
Predictions
This course is amazing
[101, 2023, 2607, 2003, 6429, 999, 102]
[-4.3630, 4.6859]
POSITIVE: 99.89%
NEGATIVE: 0.,11%

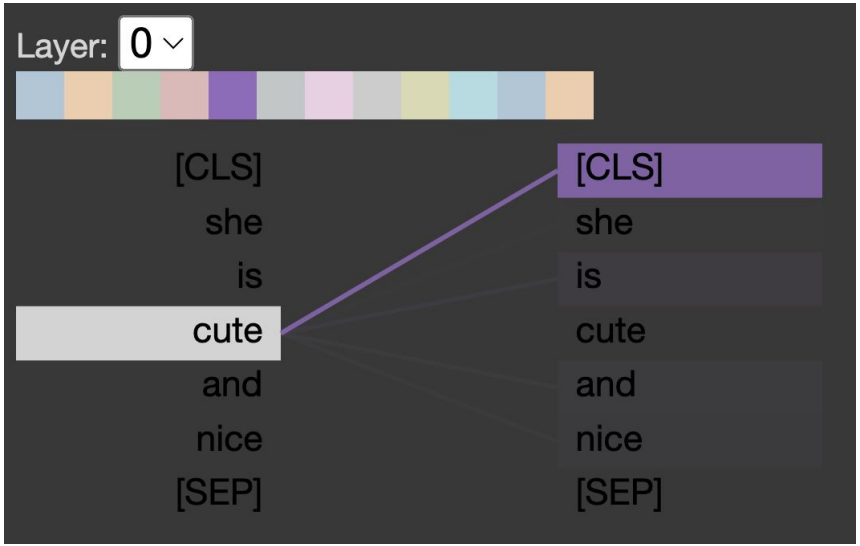
Ejemplo de embeddings con huggingface - COLAB

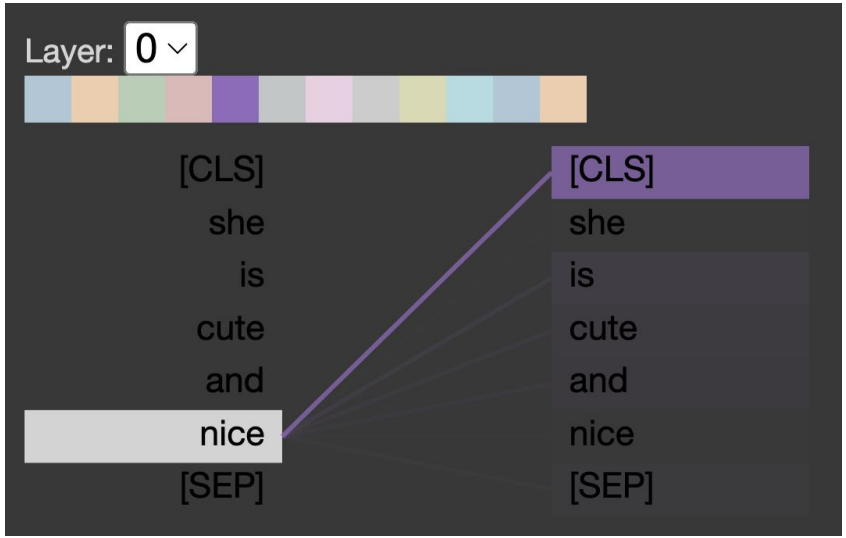


Sentiment análisis **BERT** (Paper)



Visualización self-attention en colab
Sentiment Analysis

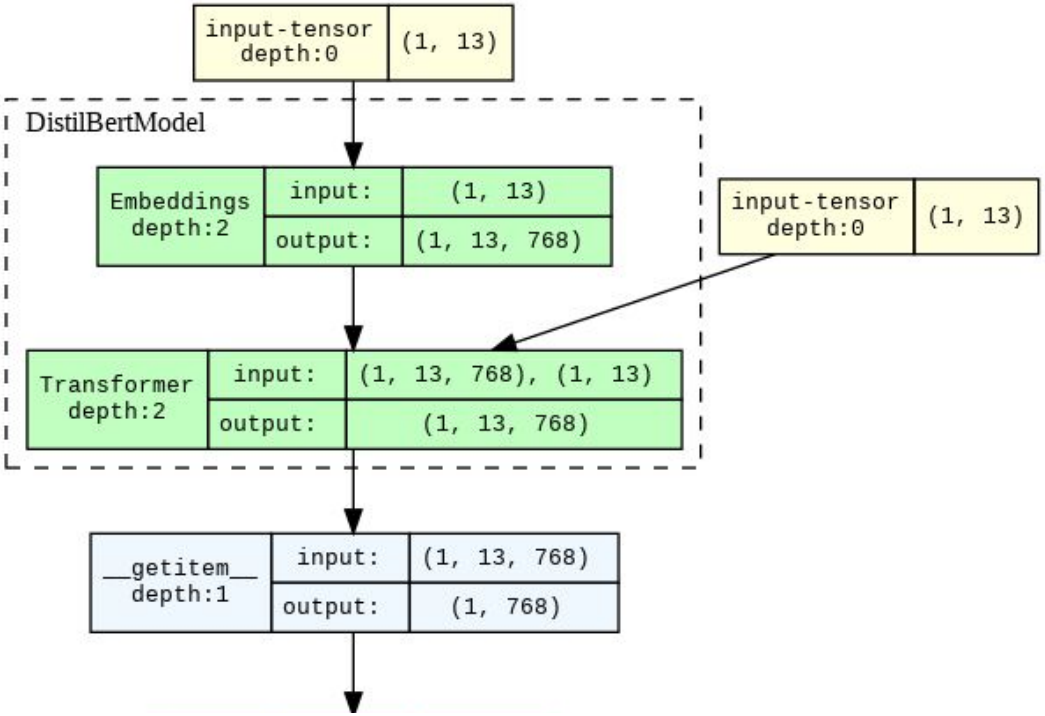


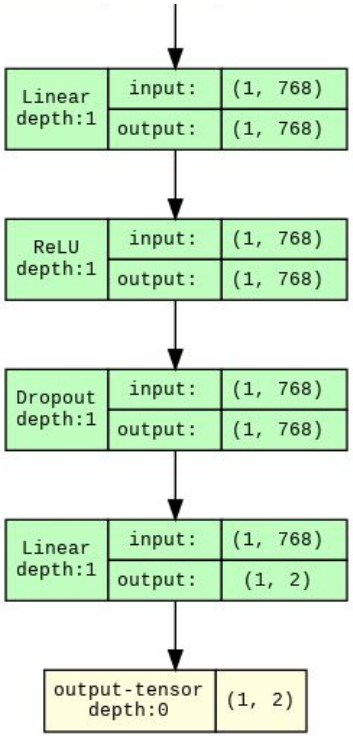


Hyper-parámetros de entrenamiento:
#capas estaqueadas = 6, #heads = 12

6x12 = 72 patrones de atención

Encoder visualization (colab)

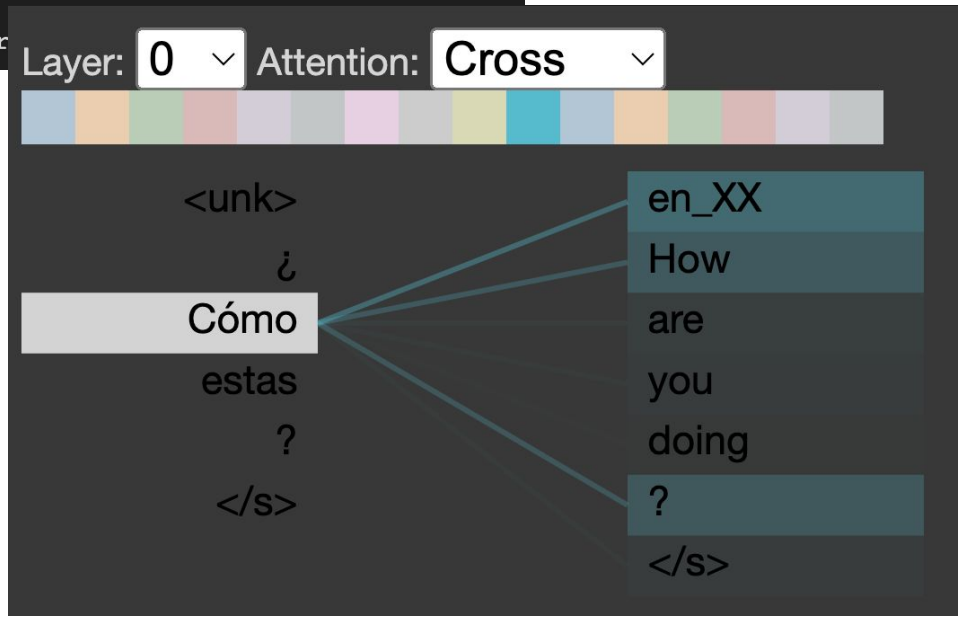




Visualización de Cross-Attention (Traducción) - colab - HF

```
src_text = " How are you doing?"
tgt_text =  "¿Cómo estas?"

model_inputs = tokenizer(src_text, return_tensors="pt")
with tokenizer.as_target_tokenizer():
    labels = tokenizer(tgt_text, return_tensor
```



Attention en decoder (GPT-2) - COLAB

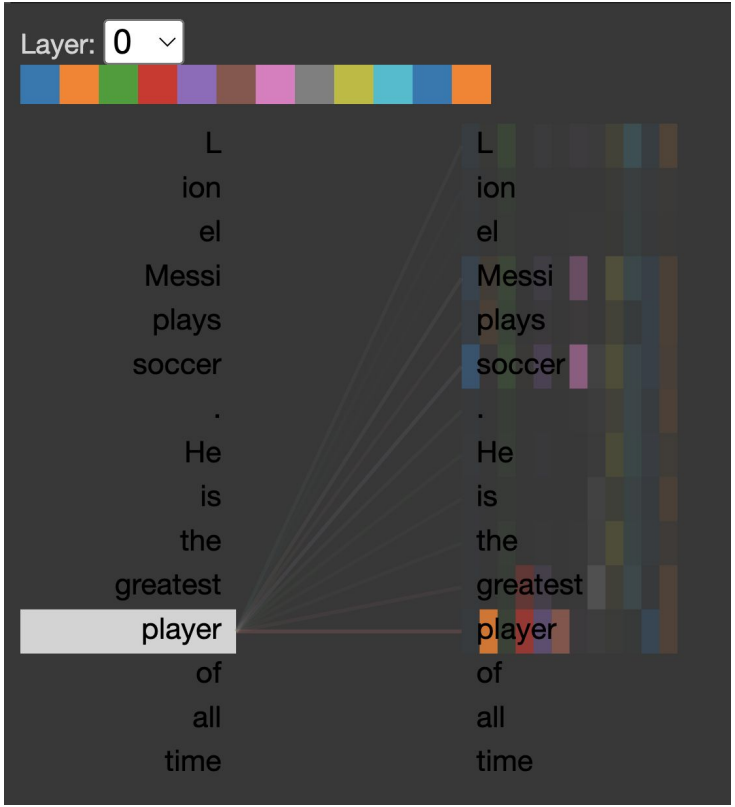
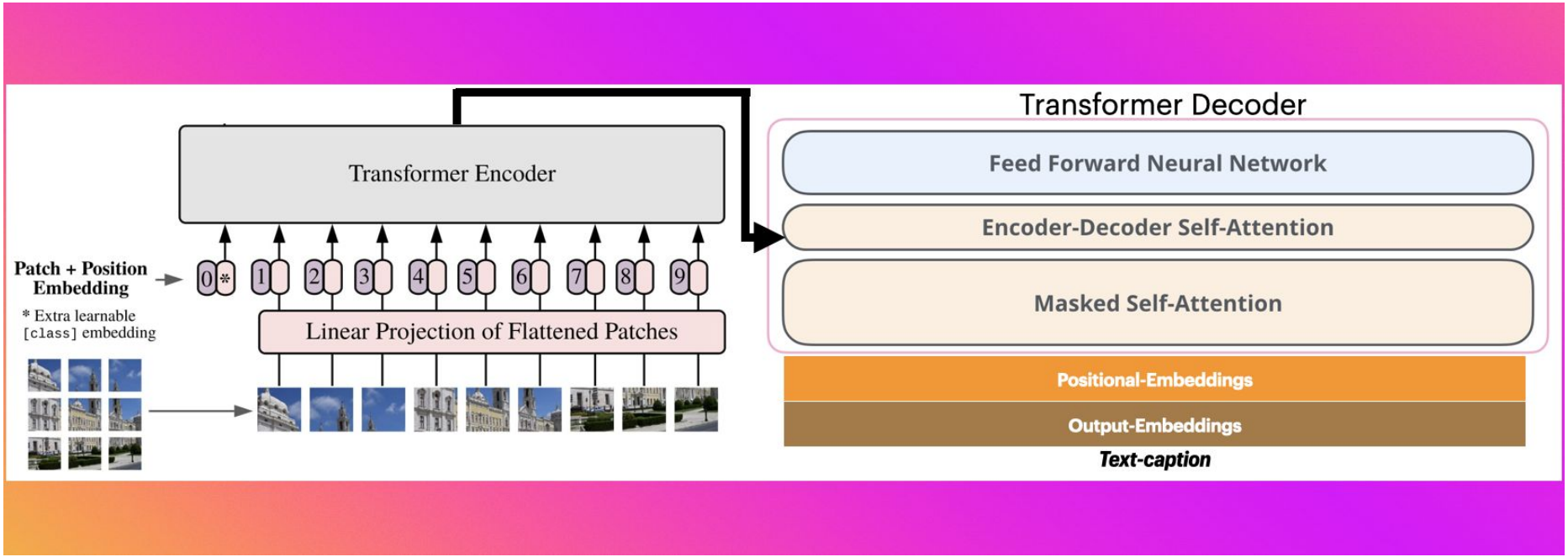
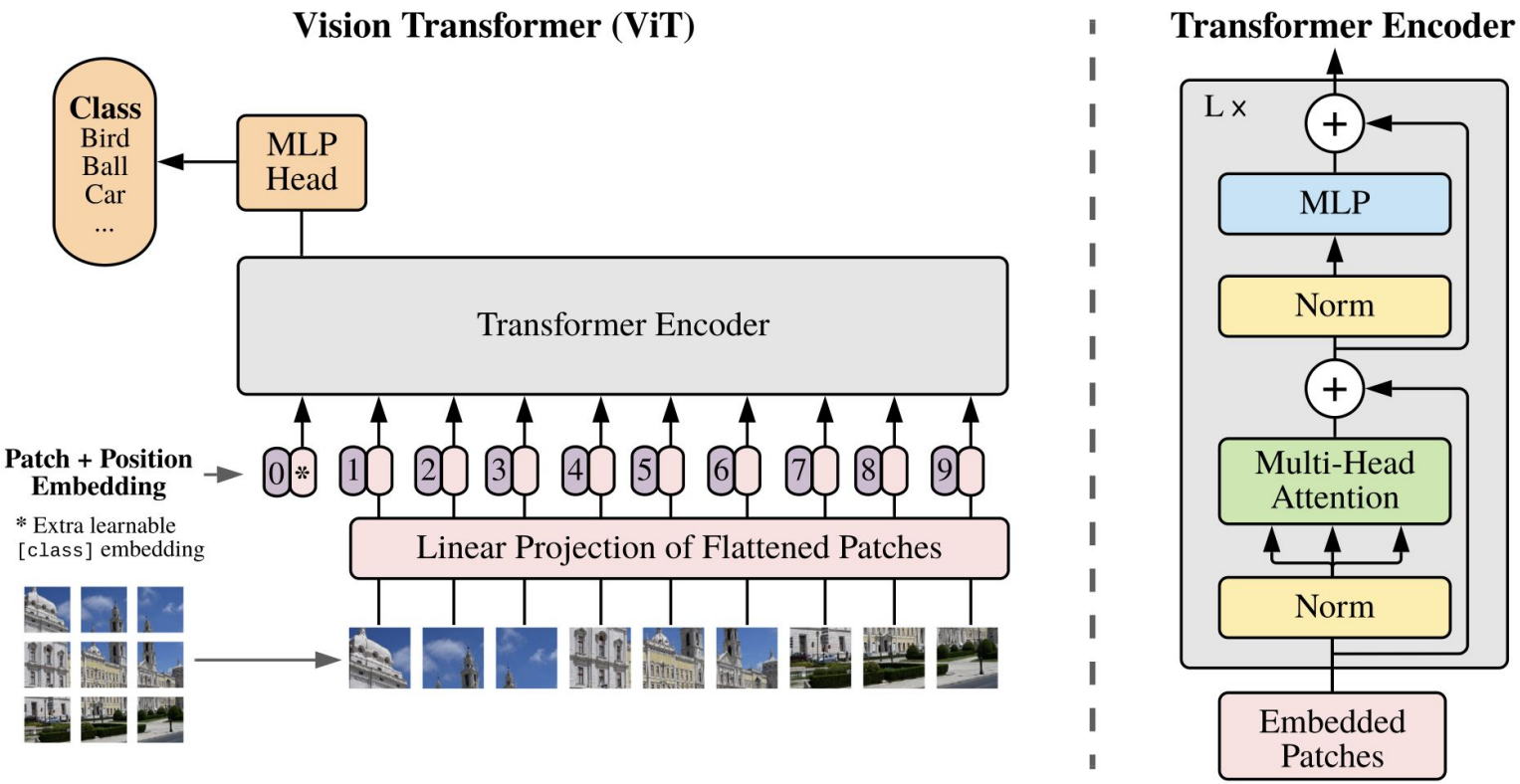


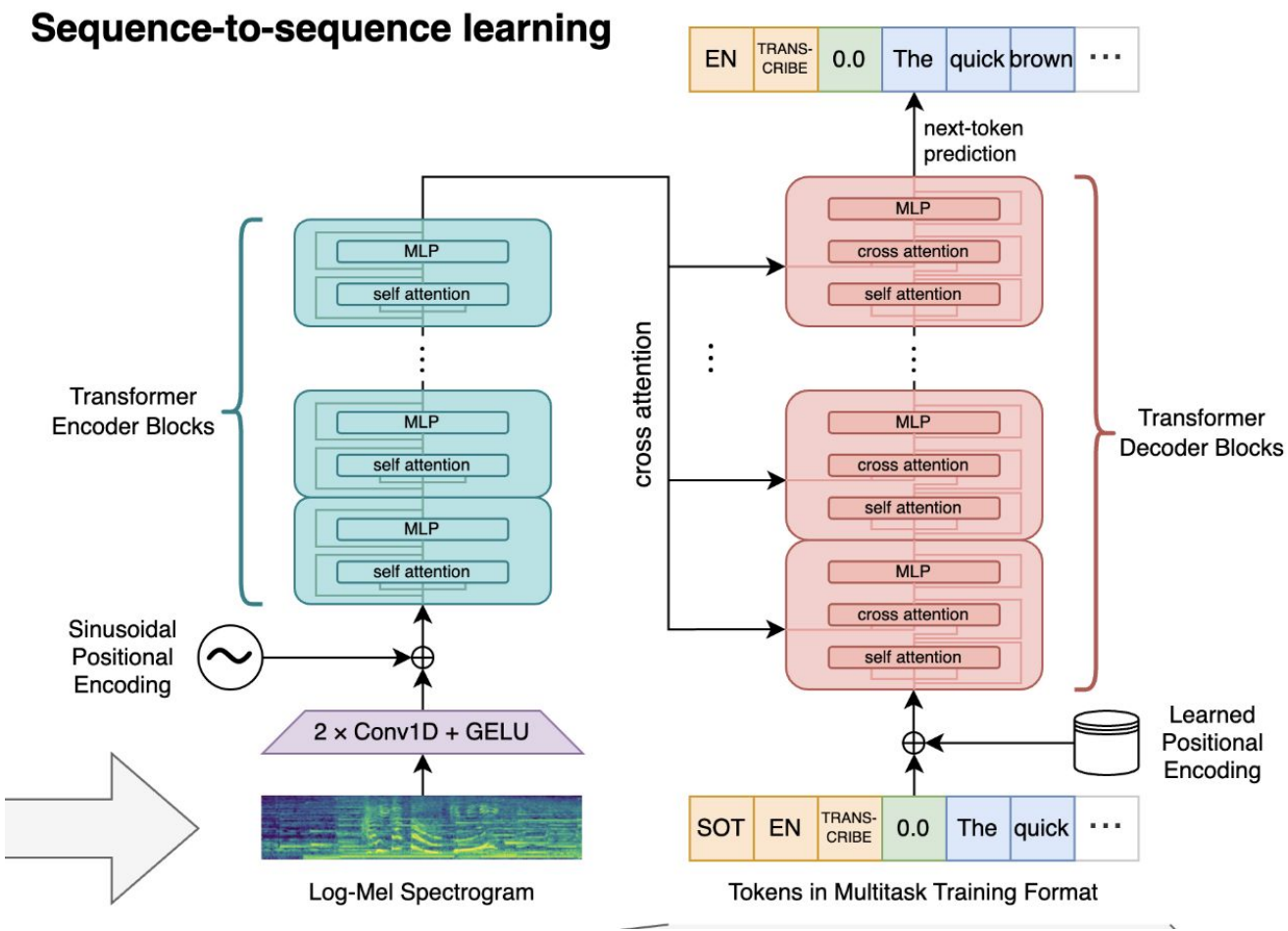
Image Captioning
Vision transformer + GTP-2 ([HuggingFace])



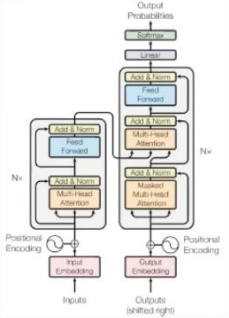
Vision Transformer



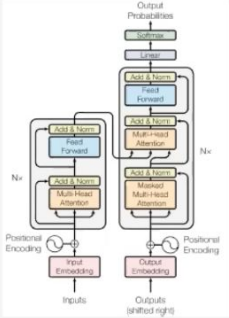
Whisper (OpenAI)



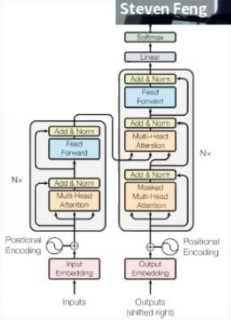
Computer Vision



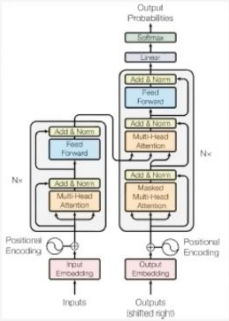
Natural Lang. Proc.



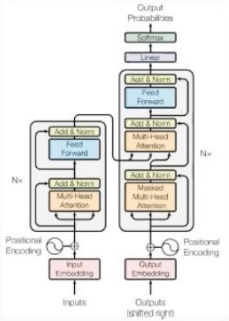
Reinf. L



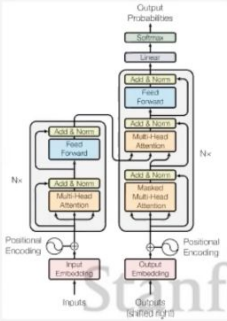
Speech



Translation



Graphs/Science



Transformer image source: "Attention Is All You Need" paper

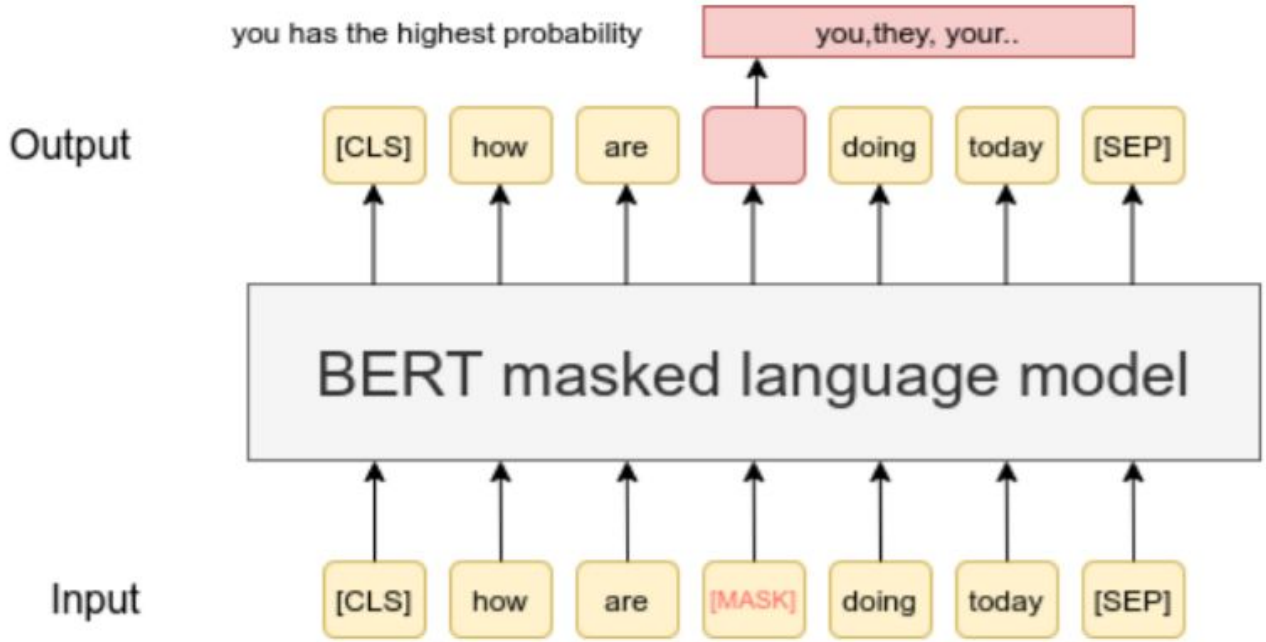
Stanford

slide from Lucas Beyer ty

Self-Supervised Training

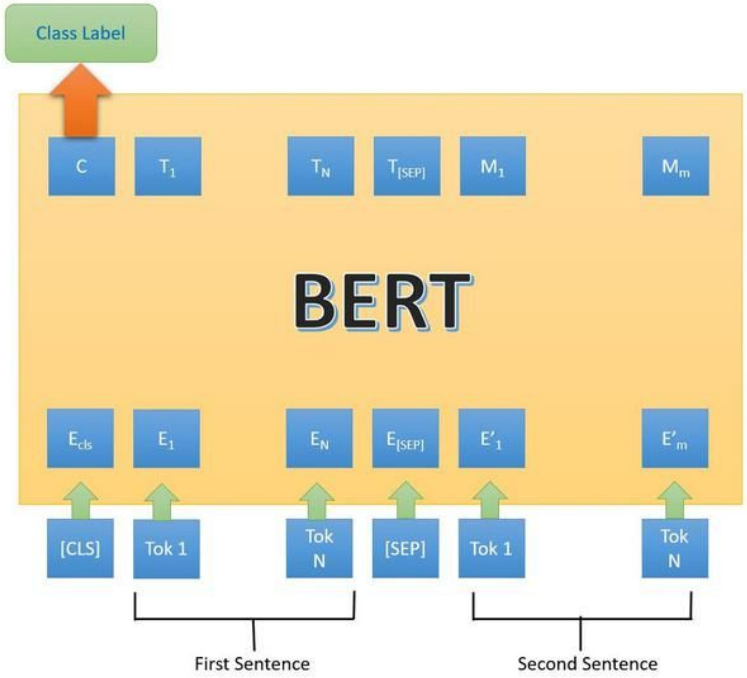
Encoder

Encoder: Entrenamiento (model card) - COLAB
Masked Language Model (MLM) - Bidireccional



BERT: Randomly masks 15% of the words.

Entrenamiento de encoder (model card)
Next Sentence Prediction (NSP)



BERT:

Input: Dos Oraciones

Output: ¿Son consecutivas?

BERT: Training and results

Training data: BookCorpus: **11,038** unpublished books, **English Wikipedia** 15% of the tokens are masked.

Método de entrenamiento:

- In 80% of the cases, the masked tokens are replaced by `[MASK]`.
- In 10% of the cases, the masked tokens are replaced by a random token (different) from the one they replace.
- In the 10% remaining cases, the masked tokens are left as is.

Preprocessing: Tokenizer: **WordPiece**, **Vocabulary:** size of **30,000**.

Tamaño de contexto: 512

#Params: 110M

#Transformers blocks: 6 **#Heads**: 12

Evaluación

Task	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	Average
	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6

MNLI: Pares de frases anotadas indicando si expresan la misma idea

QQP: Quora Question pairs

QNLI: Question Answering Natural Language Inference

SST: Stanford Sentiment Treebank

STS: Semantic Textual Similarity

Decoder

GPT

Generative Pre-Trained Transformer

GPT

Generative **Pre-Trained** Transformer

Decoder: GPT-2 - HF

Training data: To build it, **they scraped all the web pages** from outbound links on **Reddit** which received at least 3 karma. Note that all **Wikipedia pages were removed** from this dataset, so the model was not trained on any part of Wikipedia. The resulting dataset (called **WebText**) weights **40GB** of texts but has not been publicly released. You can find a list of the top 1,000 domains present in WebText here.

Método de entrenamiento: Inputs are sequences of continuous text of a certain length and the targets are the same sequence, **shifted one token** (word or piece of word) to the right. The model uses internally a mask-mechanism to make sure the predictions for the token `i` only uses the inputs from `1` to `i` but not the future tokens.

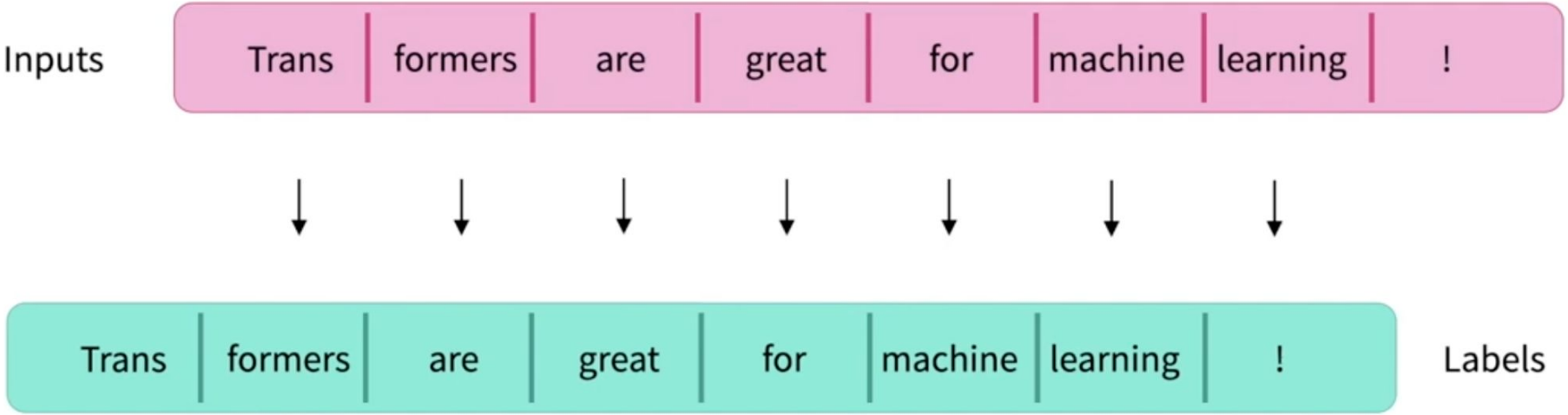
Tokenizer: byte-level version of Byte Pair Encoding (BPE), **Vocabulary:** size of **50,257**.

Tamaño de contexto: 1024

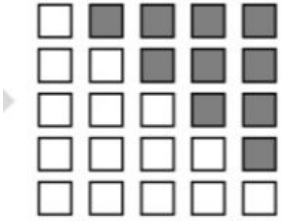
#Params: 1.5B

#Transformers blocks: 12 **#Heads**: 12

GPT2: Training



- “Trans” no se usa cómo etiqueta
- “!” no tiene etiqueta
- Con la máscara nos aseguramos que la atención solo se aplique sobre las anteriores



S_1: attend to left context

Evaluación

Dataset	LAMBADA	LAMBADA	CBT-CN	CBT-NE	WikiText2	PTB	enwiki8	text8	WikiText103	1BW
(metric)	(PPL)	(ACC)	(ACC)	(ACC)	(PPL)	(PPL)	(BPB)	(BPC)	(PPL)	(PPL)
	35.13	45.99	87.65	83.4	29.41	65.85	1.16	1,17	37.50	75.20

LAMBADA: Predicción de la última palabra
enwiki8: Los primero 100M de la wikipedia en inglés

ACC: Accuracy
PPL: Perplexity
BPB: Bits per Byte
BPC: Bits per character

GPT

Generative Pre-Trained Transformer

GPT

Generative Pre-Trained Transformer

Ejemplo GPT2 - colab

```
['L', 'ion', 'el', 'ĠMessi', 'Ġplays', 'Ġsoccer', '.', 'ĠHe', 'Ġis', 'Ġthe', 'Ġgreatest', 'Ġplayer', 'Ġof', 'Ġall', 'Ġtime']
```

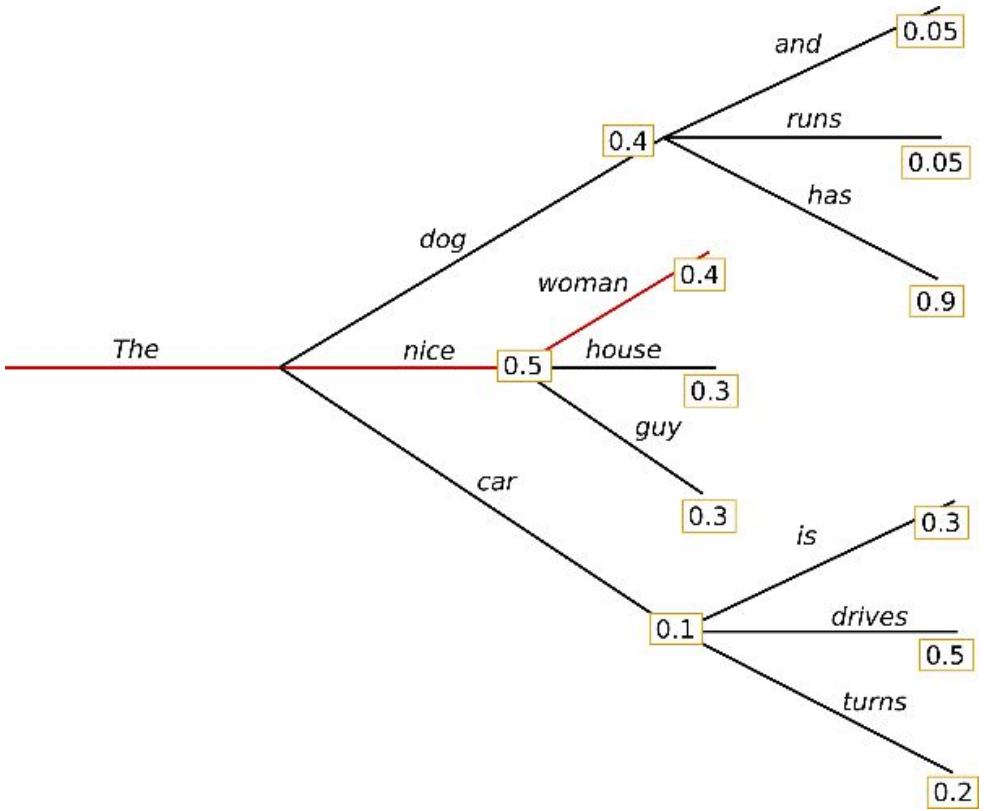
```
n = 5
print(tokens[:n])
get_top_k(outputs['logits'][0][n])
```

```
['L', 'ion', 'el', 'ĠMessi', 'Ġplays']
Ġin      0.2105
Ġat      0.1364
.        0.1002
Ġfor     0.0875
Ġwith    0.084
,        0.0587
Ġon      0.0381
Ġand     0.0365
Ġas      0.0277
Ċ        0.0222
```

```
n = 9
print(tokens[:n])
get_top_k(outputs['logits'][0][n])
```

```
['L', 'ion', 'el', 'ĠMessi', 'Ġplays', 'Ġsoccer', '.', 'ĠHe', 'Ġis']
Ġbest    0.0779
Ġmost    0.0645
Ġonly    0.0619
Ġfirst   0.0369
```

Estrategias de generación de texto COLAB - BLOG - COLAB2

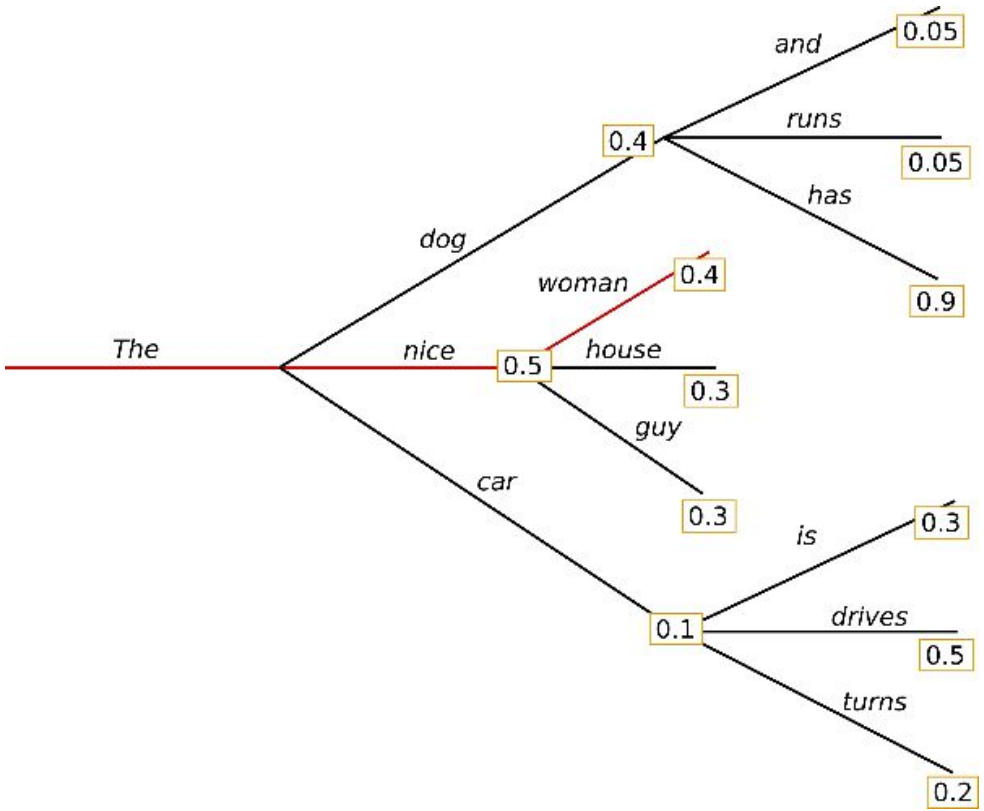


$$P(w_{1:T}|W_0) = \prod_{t=1}^{T} P(w_t|w_{1:t-1}, W_0)$$

“The nice woman” = 0.5 x 0.4 = 0.2

W_0 = “The” (contexto)
$W_{1:T}$ = “nice”, “woman”

Greedy Search

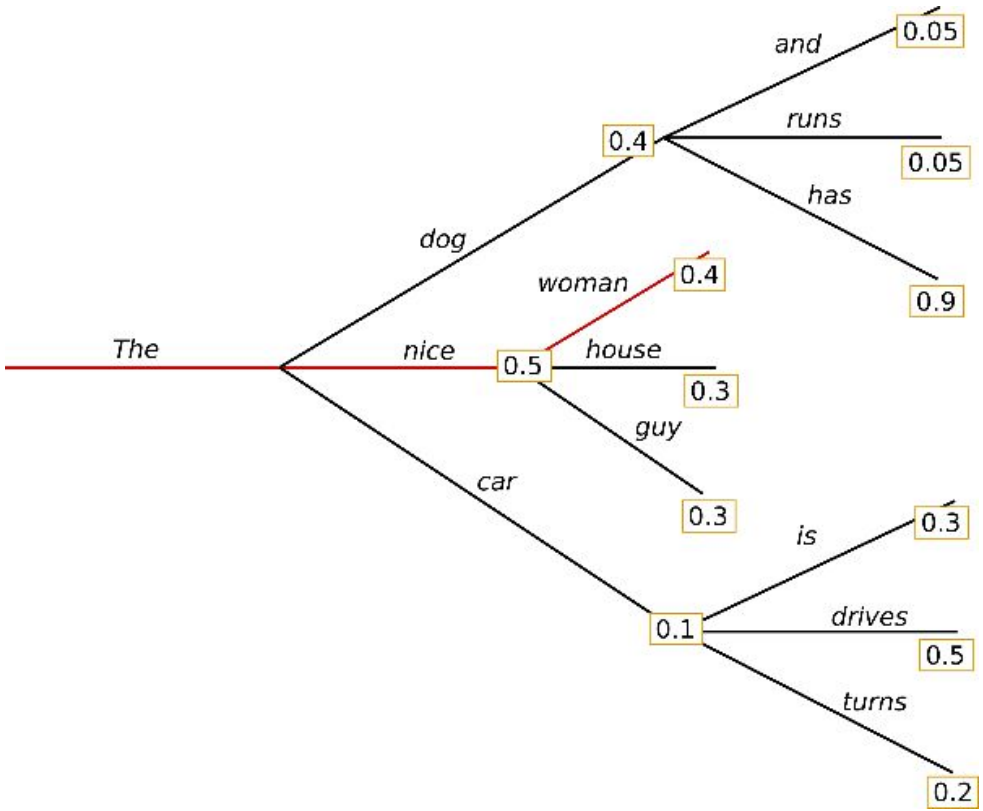


$$argmax_w P(w|w_{1:t-1})$$

Selecciono la palabra con la probabilidad más alta

¿Qué problemas tiene?

Beam Search



`num_beams=2`

Step 1:
“The nice” → 0.5
“The dog” → 0.4

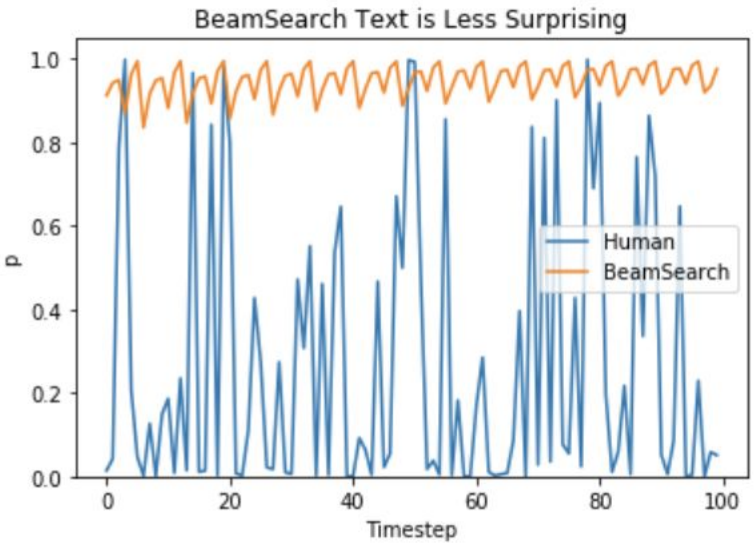
Step 2:
“The nice woman” → 0.5x0.4 = 0.2
“The dog has” → 0.4x0.9 = 0.36

Siempre encuentra una secuencia con prob >= que Greedy Search

No asegura que encuentra la más probable

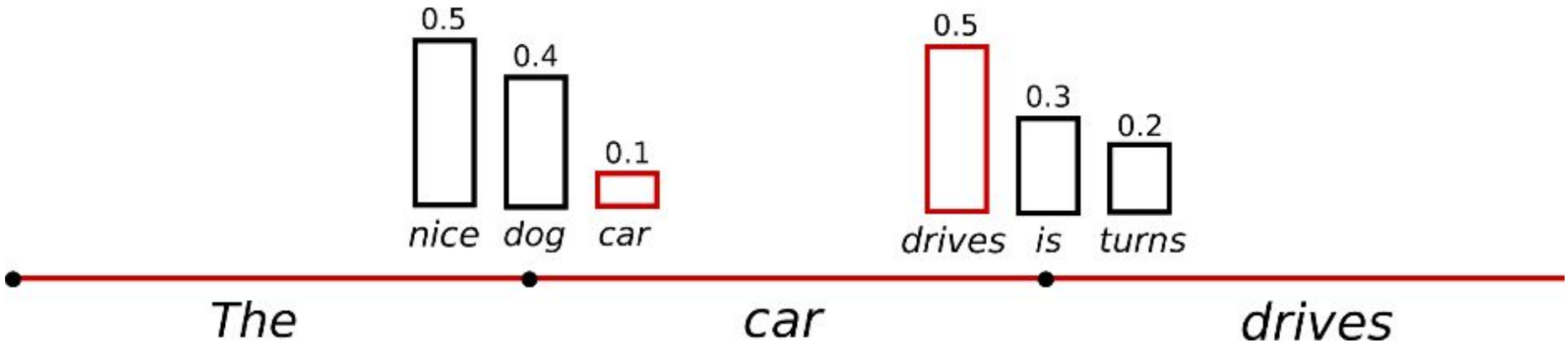
Conclusiones Beam Search

- Funciona mejor cuando la longitud de la secuencia es más o menos predecible → Traducción o sumarización
- No tan bien para generación de texto donde el final es abierto (Diálogos o historias)
- Se puede evitar la repetición usando n-gram pero hay que tener cuidado y no es fácil de ajustar.
- Textos interesantes escritos por humanos no siguen una distribución de alta probabilidad para la próxima palabra. (Sorpresa, no previsible)



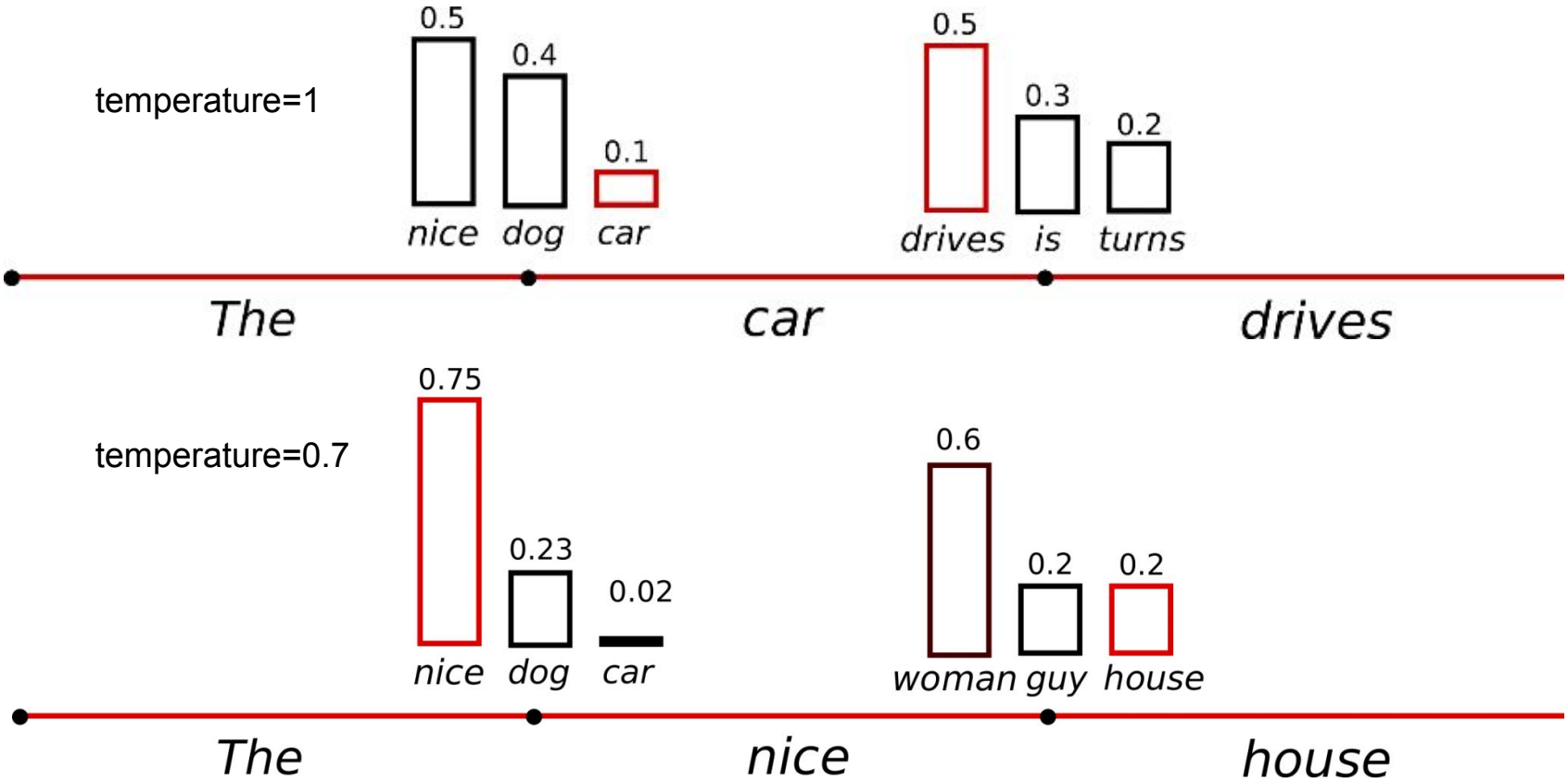
Muestreo (Sampling)

$$w_t \sim P(w|w_{1:t-1})$$

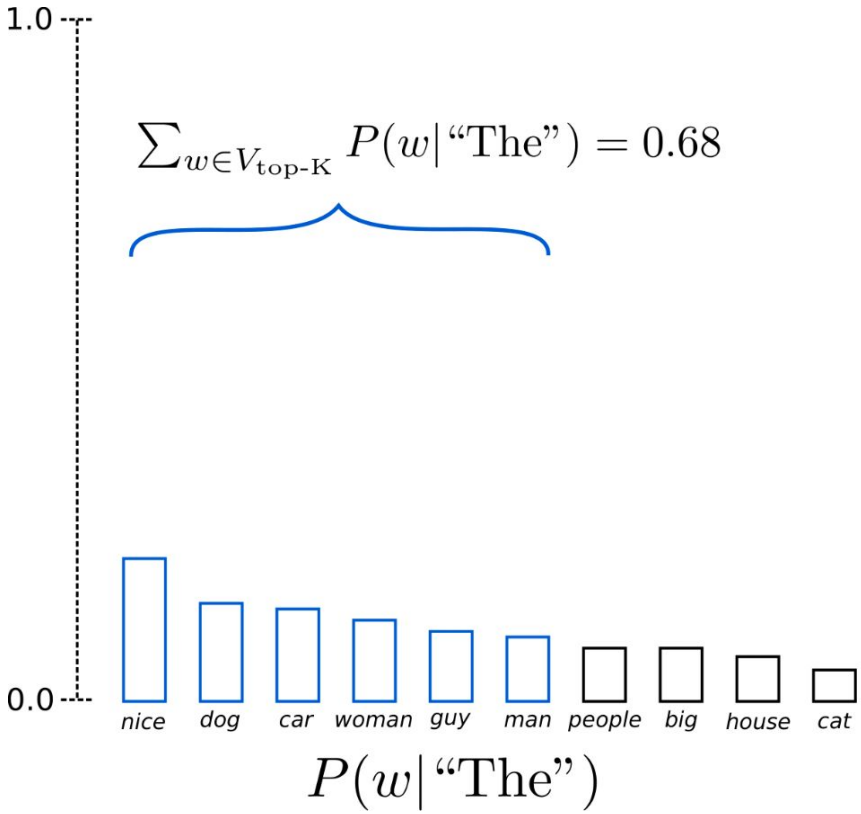


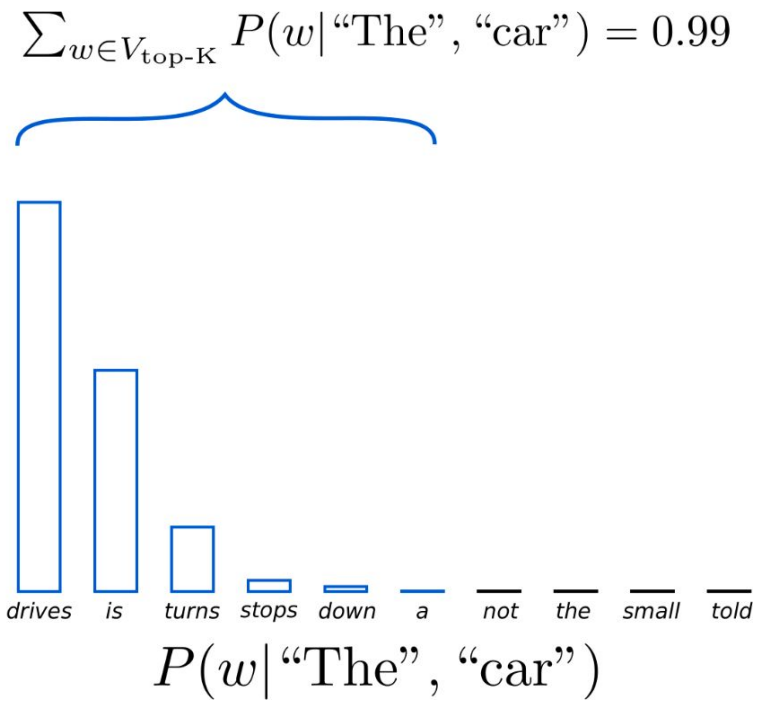
temperature → 0 = Greedy Search

Temperatura



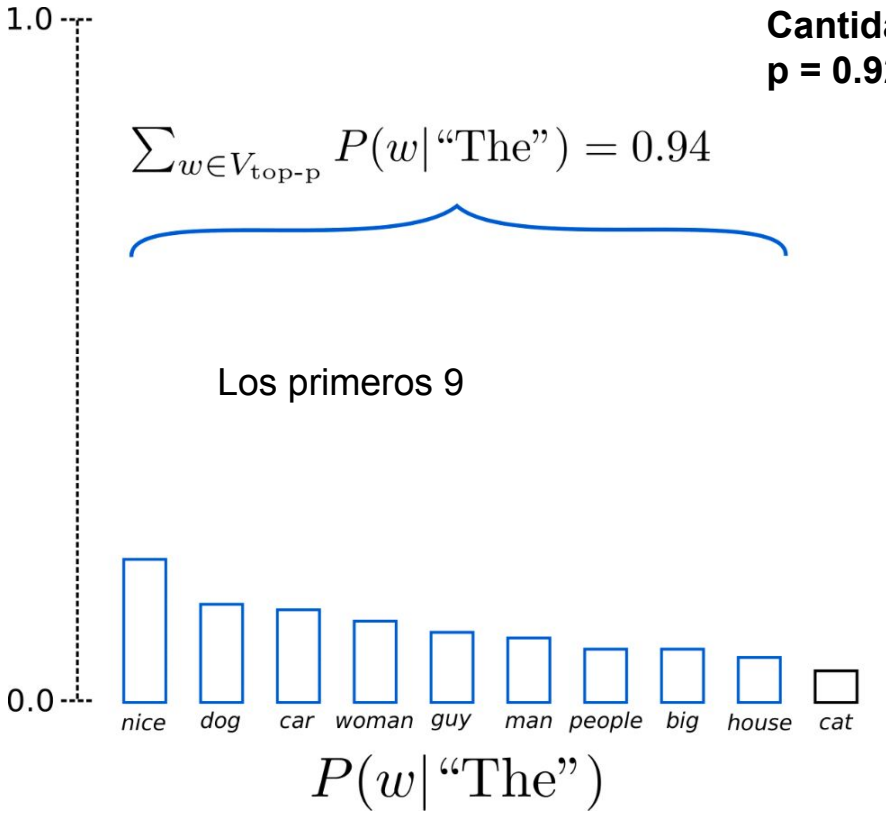
Top-k sampling

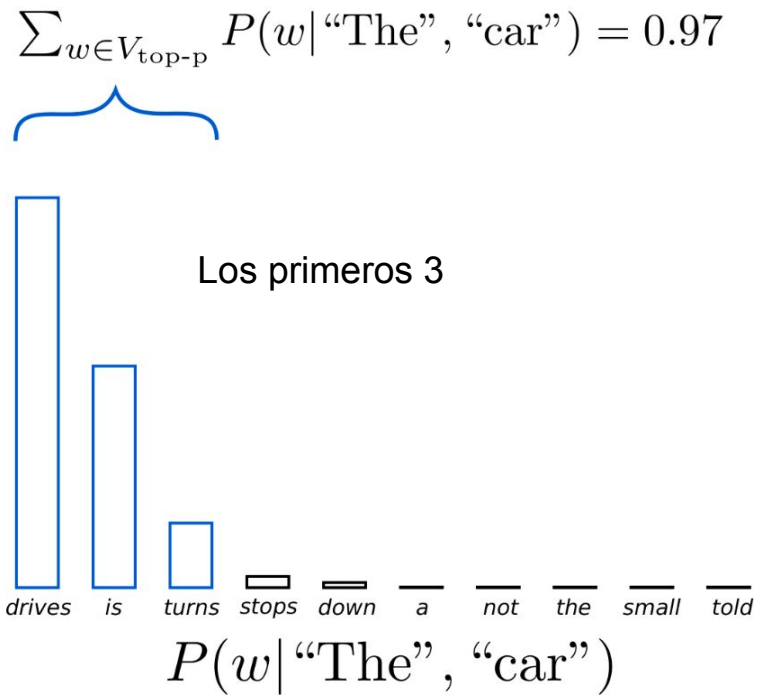




Top-p nucleus sampling

Cantidad mínima que exceda
p = 0.92





BLIP: Image Captioning (COLAB)

Resumen: Generación de texto

Técnicas

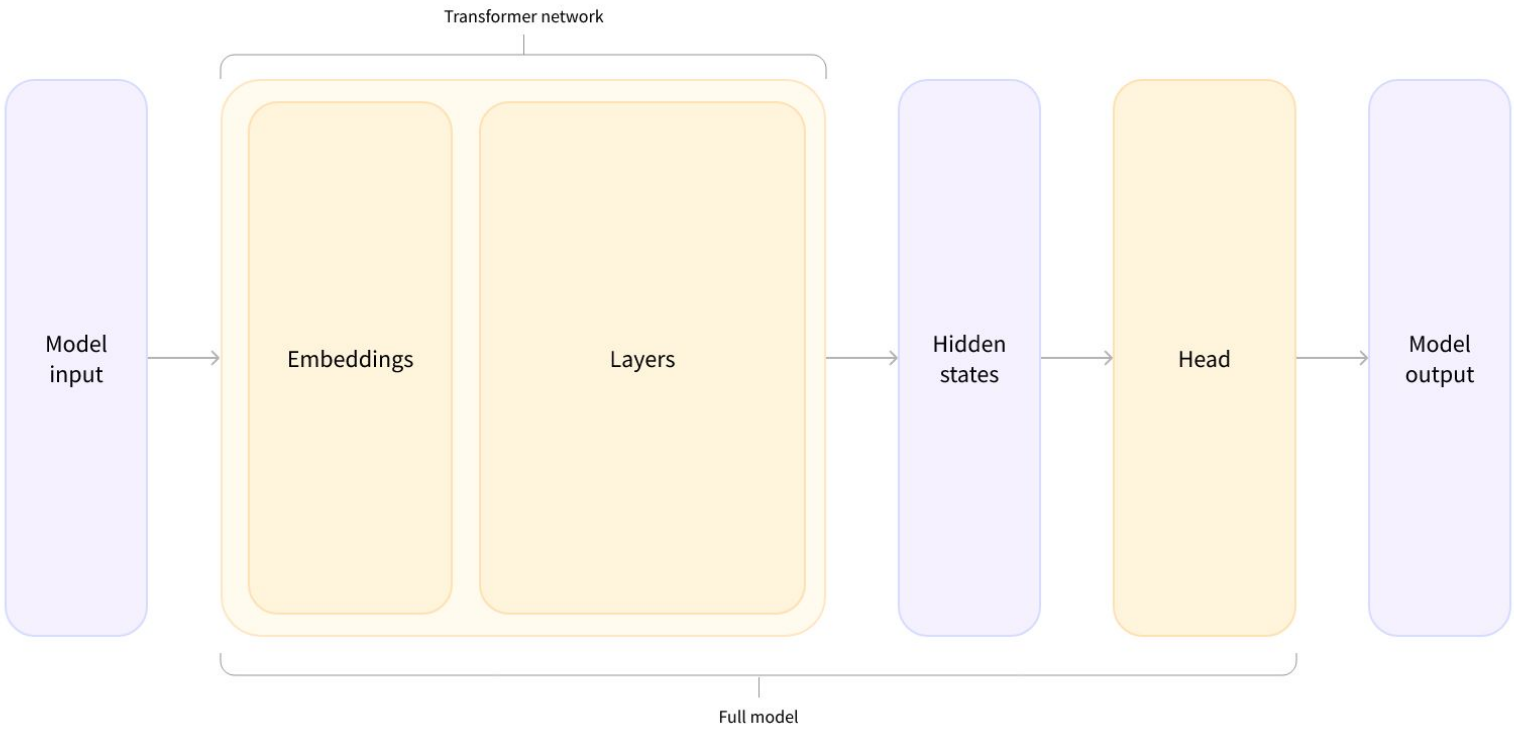
- Greedy Search
- Beam Search
- Muestreo
- Top-k sampling
- Top-p nucleus sampling
- Stochastic BS = Beam Search + Muestreo

Parámetros

- num_beams
- no_repeat_ngram_size
- do_sample
- top_k
- temperature
- top_p
- repetition_penalty
- min_length

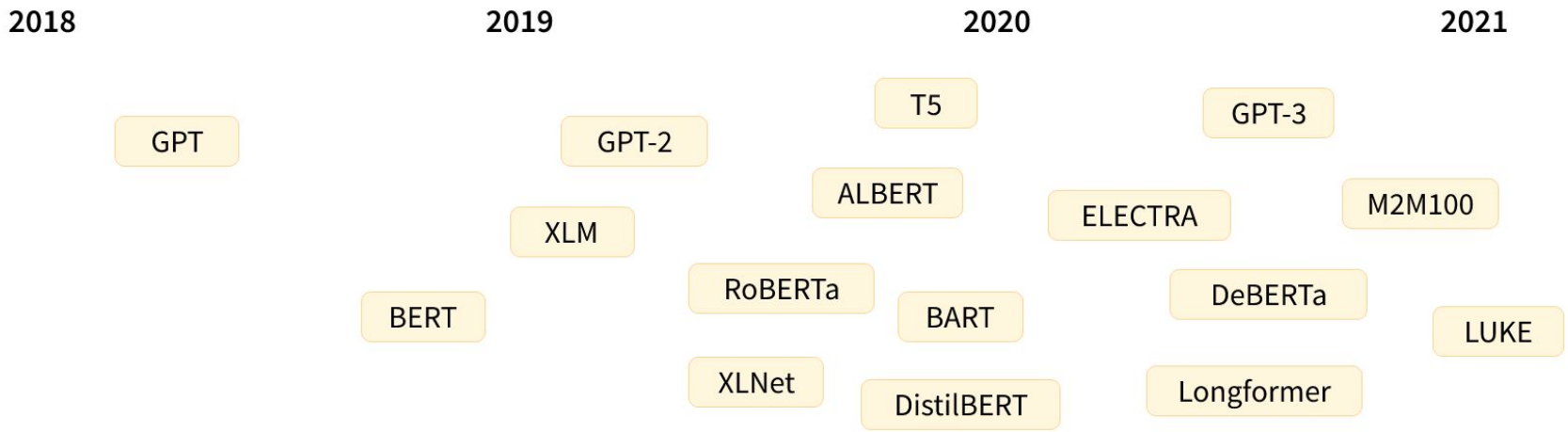
top-p y top-k se pueden combinar

Fine-tuning


Transformer network
Model input
Embeddings
Layers
Hidden states
Head
Model output
Full model

Resumen y generalidades

Transformers



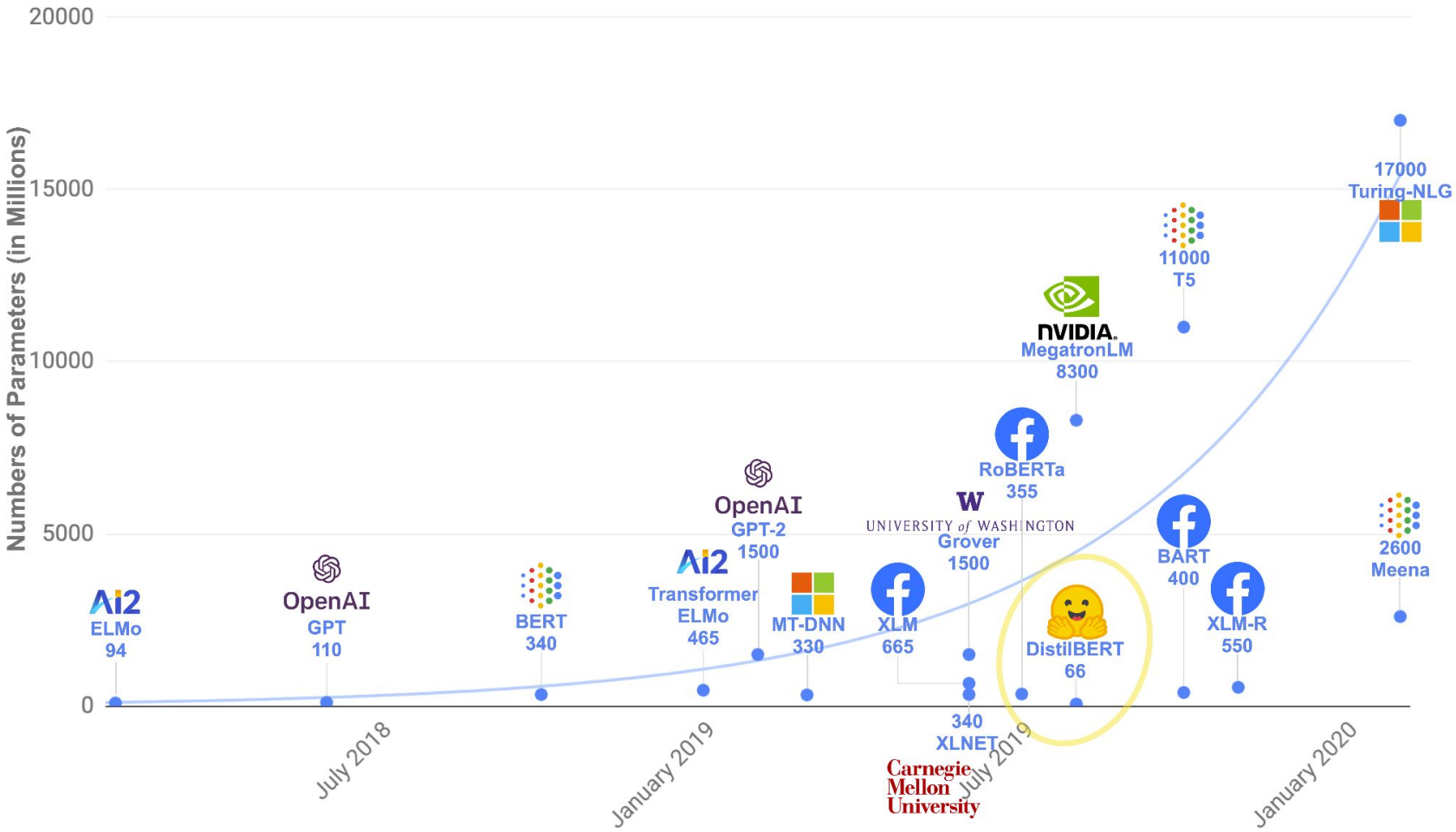
- Introducidos en 2017 para resolver el problema de traducción (paper)
- 2018 (Jun): GPT, Primer transformer **pre-entrenado** utilizado para **varias tareas** de **NLP**
- 2018 (Oct): BERT, focalizado en realizar resúmenes
- 2019 (Feb): GPT-2, versión mejorada (más grande) de GPT
- 2019 (Oct): DistilBERT, 60% más rápida y 40% más liviana en memoria que BERT (97% BERT perf)
- 2019 (Oct): BART y T5, Misma arquitectura que el original pero más grande (Pre entrenada)
- 2020 (May): GPT-3, versión mejorada (más grande) de GPT-3. Excelente performance para varias tareas sin fine-tuning (zero shot learning)

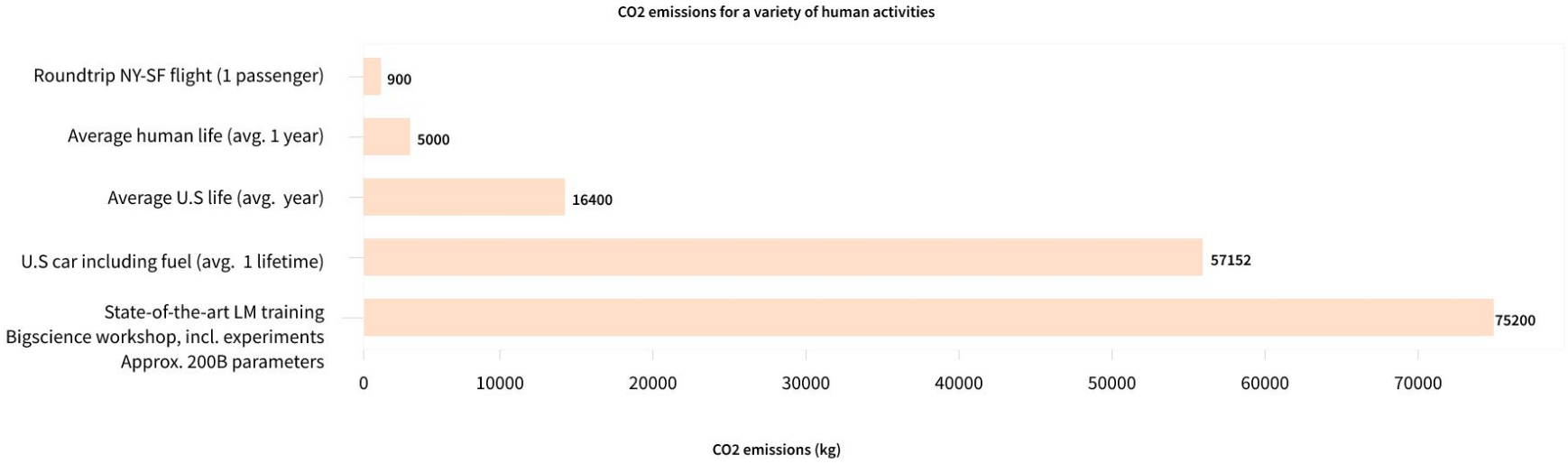
Clasificación de los transformers

- GPT-like
  - Auto-regresivos
- BERT-like
  - Auto-encoding
- BART/T5-like
  - Sequence-to-sequence

Aprendizaje Auto-supervisado: es un término que se refiere a un tipo de aprendizaje no supervisado enmarcado dentro de un problema de aprendizaje supervisado

- Son todos modelos de lenguaje entrenados de forma auto-supervisada (self-supervised)
- No se precisan seres humanos para anotar la data


Numbers of Parameters (in Millions)
20000
15000
10000
5000
0
July 2018
January 2019
July 2019
January 2020
Ai2
ELMo
94
OpenAI
GPT
110
BERT
340
Ai2
Transformer
ELMo
465
OpenAI
GPT-2
1500
MT-DNN
330
XLM
665
340
XLNET
Carnegie Mellon University
W
UNIVERSITY of WASHINGTON
Grover
1500
RoBERTa
355
NVIDIA
MegatronLM
8300
DistilBERT
66
BART
400
XLM-R
550
11000
T5
17000
Turing-NLG
2600
Meena


CO2 emissions for a variety of human activities
Roundtrip NY-SF flight (1 passenger)
900
Average human life (avg. 1 year)
5000
Average U.S life (avg. year)
16400
U.S car including fuel (avg. 1 lifetime)
57152
State-of-the-art LM training
Bigscience workshop, incl. experiments
Approx. 200B parameters
75200
0
10000
20000
30000
40000
50000
60000
70000
CO2 emissions (kg)

Referencias

- All you need is transformers: 2017 paper
- Andrej Karpathy Stanford Seminar Video
- A Neural probabilistic Language model
- The illustrated Transformer
- NMT by Jointly learning to align and translate
- Sequence to Sequence Learning with Neural Nets
- http://cs231n.stanford.edu/slides/2022/lecture_11_ruohan.pdf
- https://huggingface.co/blog/how-to-generate