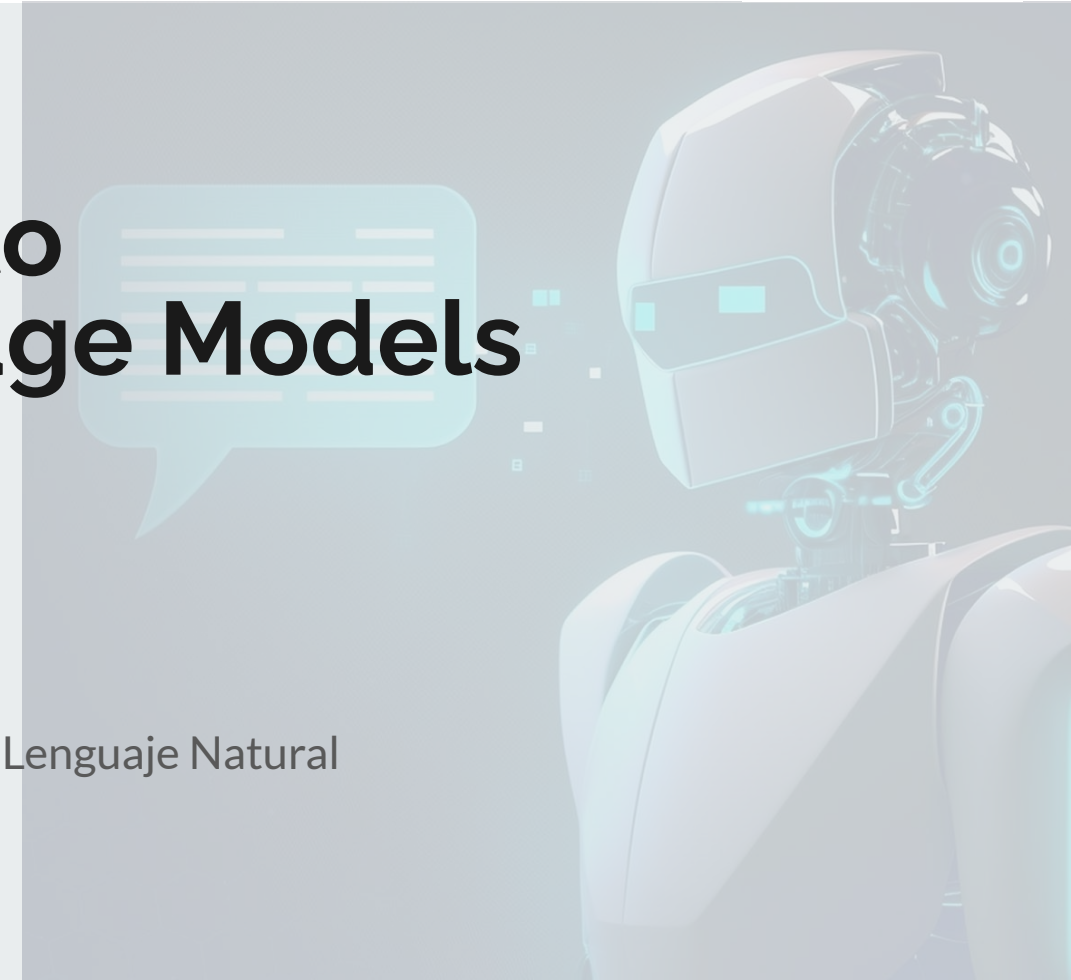
Introduction to Large Language Models and Agents



Ronald Cardenas Acosta, Ph.D.
Investigador en Procesamiento de Lenguaje Natural

Syllabus

Unidad 1	Conceptos Básicos
Unidad 2	Modelado de Lenguaje
Unidad 3	**Pre-Entrenamiento y Fine-tuning**
Unidad 4	Post-entrenamiento, aplicaciones de LLMs

Unidad 3: Pre-Entrenamiento Y Finetuning

Pretraining

Prompting

Leyes de Escalamiento

Métodos de Generación de Texto

Generación de Texto: Demo

Pretraining

Tokenizacion

Pretraining vs Finetuning

Pretraining Encoders
- *Masked Language Models*
- BERT

Pretraining Encoder-Decoders
- BART, T5, T0

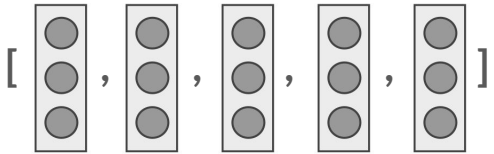
Pretraining Decoders
- GPT family

Tokenización: Unidades sub-palabra

"Ellos jugaban basketball"

["Ellos", "jug", "#aban", "basket", "#ball"]

[32, 5, 1856, 957, 646]



Separamos en unidades sub-palabra (wordpieces)

- Estas piezas conforman vocabulario
- Palabra = 1 o más piezas
- Cómo se obtienen estas piezas? Byte-pair encoding*

Mapeamos cada unidad a su ID en el vocabulario

Obtenemos el embedding de cada unidad

(*) Sennrich, Rico, Barry Haddow, and Alexandra Birch. "Neural Machine Translation of Rare Words with Subword Units." *Proceedings of the ACL 2016 (Volume 1: Long Papers)*. Association for Computational Linguistics, 2016.

Pre-entrenamiento de Transformers

Limitacion

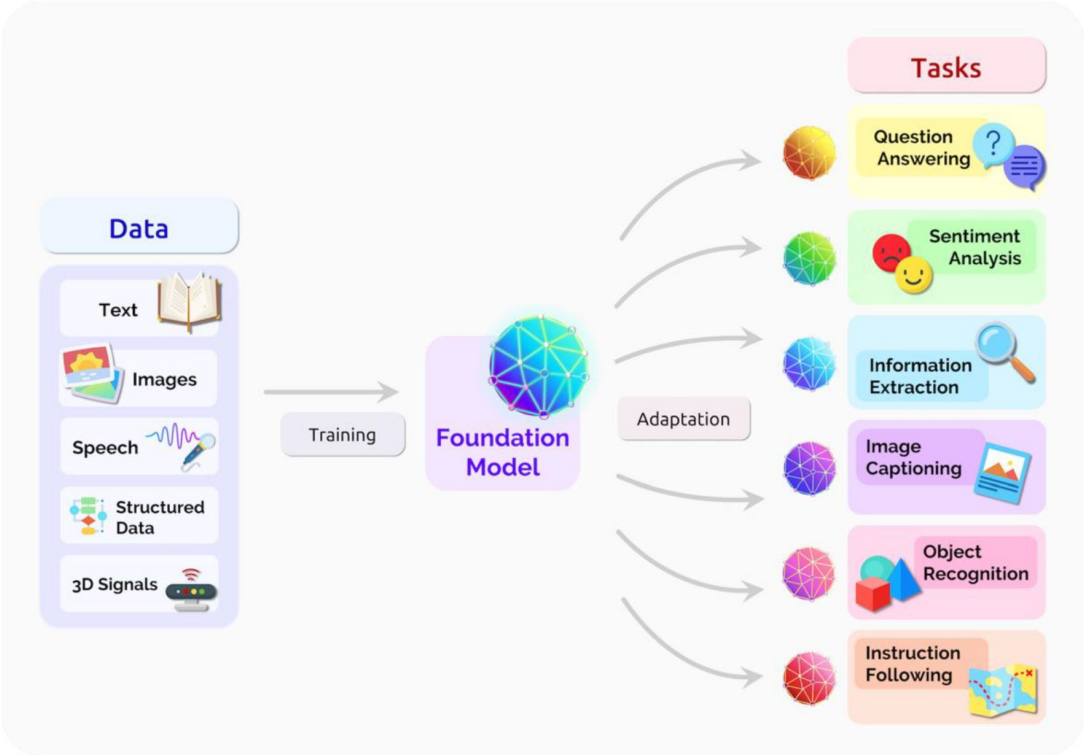
- Transformers requieren de grandes cantidades de data de entrenamiento
- Data para entrenamiento supervisado ***siempre será limitada***

Solucion (parcial)

- Usar data sin labels, e.j. texto raso del internet
- Diseñar una tarea que use partes de la misma data como entrada (**x**) y salida (**y**)
  - *Modelado de lenguaje es perfecto para esto !*
  - “Self-supervised” training

Next-token Prediction: “a b c” -> x, y =(‘a b’, ‘c’) P(c | a b)

Pre-entrenamiento y Finetuning



Pre-entrenamiento y Finetuning

Pretraining / Pre-entrenamiento

- Primera fase de entrenamiento de un transformer
- Requiere de bastante data y FLOPs (compute)
- Se usa tareas de self-supervision (ej next token prediction)
- Objetivo:
    - Entrenar el modelo a codificar los patrones generales de la data (texto)

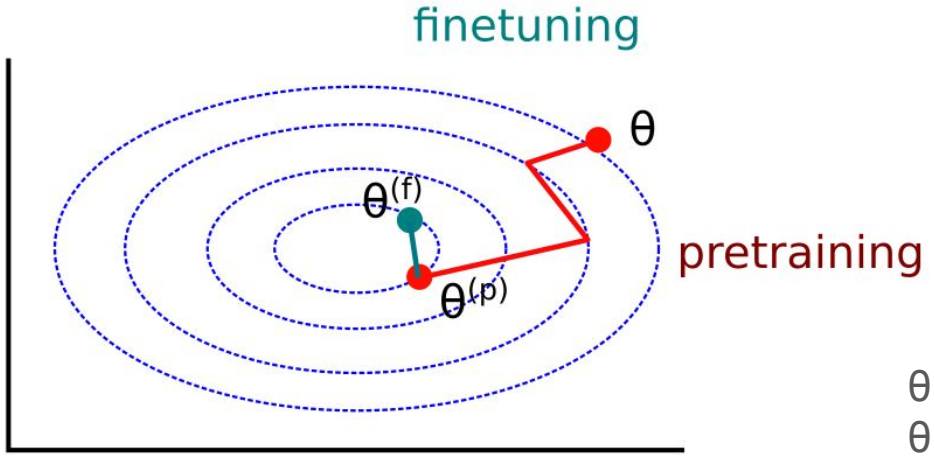
Finetuning

- Entrenamiento en una tarea en específico (ej. Sentiment Analysis)
- Require data supervisada, y mucho menos FLOPs

"Pre-entrena una vez, fine-tune muchas veces"

Pre-entrenamiento: porque funcionaria?

- Util como inicializador de parametros
- Desde una perspectiva de optimización:



Llegar a θ^(f) es mas facil si partimos desde θ^(p) que si partimos de θ

θ : parametros iniciales
θ^(p): parametros luego de pretraining
θ^(f) : parametros luego de finetuning

Pretraining data: Question Answering vs Texto raso

- Datasets de Question Answering (QA): *calidad acceptable*
- Datasets de texto raso del internet:
  - ruidoso, requiere bastante filtrado

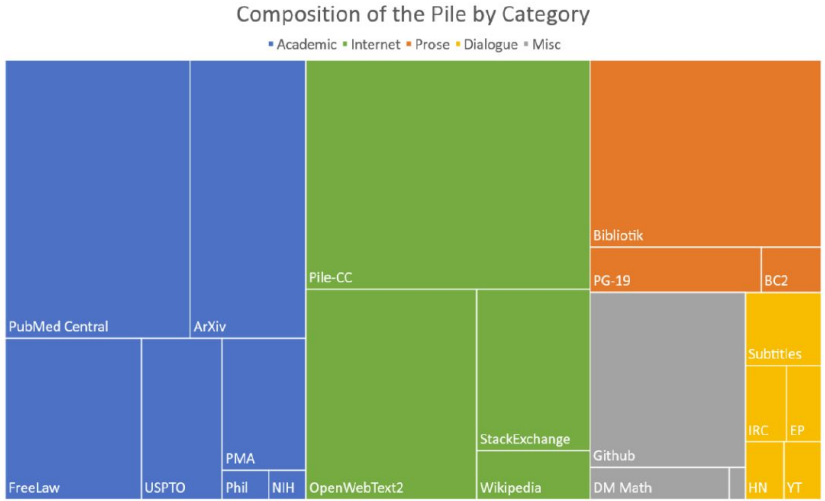
Pero…

(1) Órdenes de magnitud de diferencia en tamaño

Dataset	Tokens (~0.75 words)
SQuAD 2.0 [Rajpukar+ 2018]	< 50 Million
DCLM-pool [Li+ 2024]	240 Trillion
Estimated ‘internet text’ [Villalobos 2024]	510T (indexed), 3100T (total)

Pretraining data: Texto raso vs Question Answering

(1) Órdenes de magnitud de diferencia en tamaño
(2) Diversidad de topicos



[Gao+ 20]

Source	Doc Type	UTF-8 bytes *(GB)*	Documents *(millions)*	Unicode words *(billions)*	Llama tokens *(billions)*
Common Crawl	web pages	9,812	3,734	1,928	2,479
GitHub	code	1,043	210	260	411
Reddit	social media	339	377	72	89
Semantic Scholar	papers	268	38.8	50	70
Project Gutenberg	books	20.4	0.056	4.0	6.0
Wikipedia, Wikibooks	encyclopedic	16.2	6.2	3.7	4.3
Total		**11,519**	**4,367**	**2,318**	**3,059**

[Soldani+ 24]

Permisos de uso, atribucion, otros dilemas

BUSINESS

'The New York Times' takes OpenAI to court. ChatGPT's future could be on the line

UPDATED JANUARY 14, 2025 · 4:27 PM ET

Bobby Allyn

A sign for The New York Times hangs above the entrance to its building, Thursday, May 6, 2021, in New York. The New York Times filed a federal lawsuit against OpenAI and Microsoft on Wednesday, Dec. 27, 2023, seeking to end the practice of using published material to train chatbots.

Mark Lennihan/AP Photo

Artists Score Major Win in Copyright Case Against AI Art Generators

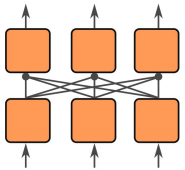
The court declined to dismiss copyright infringement claims against the AI companies. The order could implicate other firms that used Stable Diffusion, the AI model at issue in the case.

BY WINSTON CHO AUGUST 13, 2024 1:09PM

BORIS SV / GETTY IMAGES

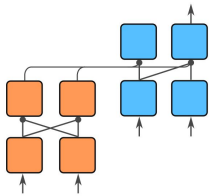
Pre-entrenamiento: Arquitecturas

El tipo de pretraining depende de la arquitectura y de las tareas de aplicación.



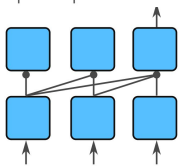
Encoder

- Contexto bidirectional, condiciona en tokens futuros
- Adecuados para representación robusta de texto



Encoder-Decoder

- Representación y generación tienen parámetros dedicados
- Adecuados cuando la secuencia source y target difieren en varios aspectos (e.g. lenguaje, estilo, formato, etc)



Decoder

- Representación y generación comparten parámetros
- Eficientes para generación auto-regresiva (una palabra a la vez)

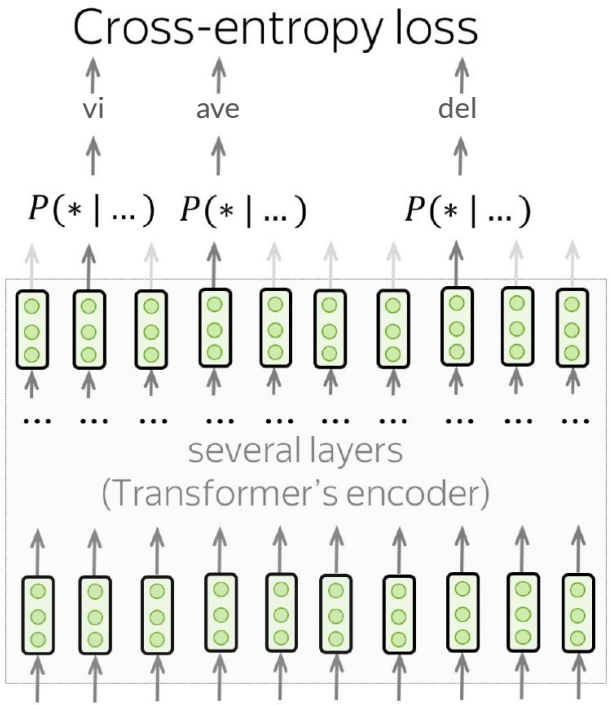
Pretraining Encoders

Pre-entrenando Encoders

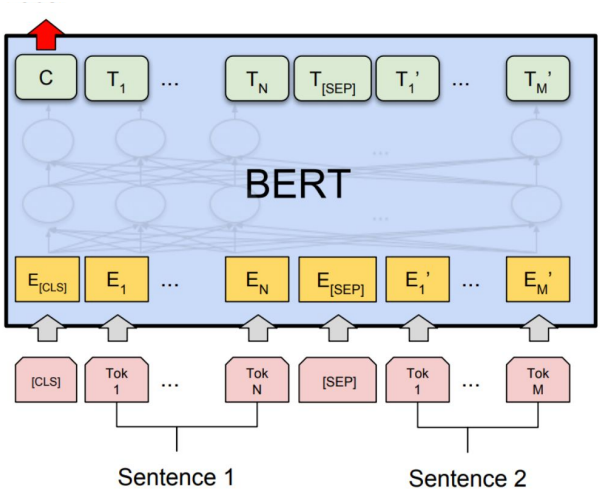
Contexto bidireccional, no es posible hacer LM con next-token-prediction

Solucion: *Masked* Language Modeling

- Reemplaza aleatoreamente X% de palabras con tag [MASK], predice esas palabras
- Calcula CE loss **sólo** para palabras reemplazadas



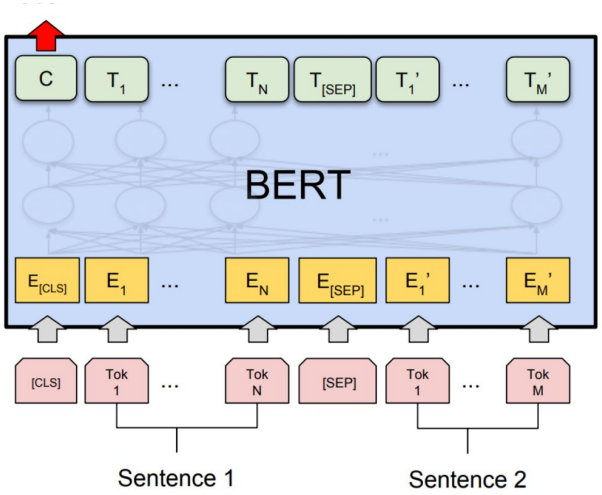
BERT: Bidirectional Encoder Representations from Transformers



- "Bidirectional"
  - usa multi-head ***self***-attention
  - en vez de ***causal*** attention
- Diseñado para tareas de clasificación de texto
- Pre-entrenado
  - *Masked Language Modeling*
  - *Next-Sentence-Prediction*

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *Proceedings of NAACL 2019, volume 1 (long and short papers)*. 2019.

BERT: Bidirectional Encoder Representations from Transformers

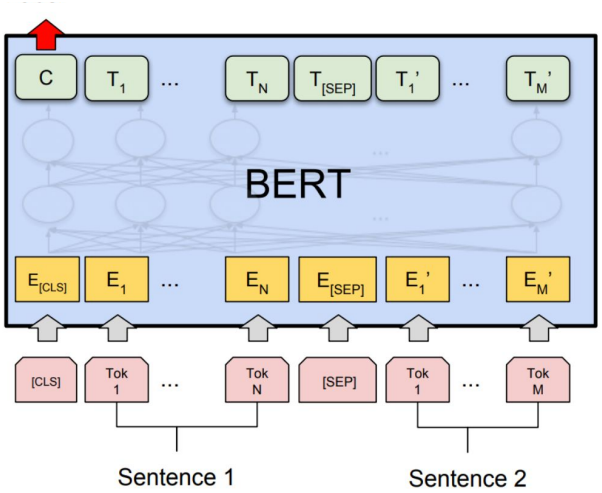


Masked Language Modeling

- Selecciona 15% de tokens (wordpieces) para predicción
  - Reemplaza token con [MASK] con prob. 80%
  - Reemplaza token con palabra aleatorea con prob. 10%
  - Dejar token intacto con prob. 10% (pero predecirlo igual)
- Fuerza al modelo a aprender representaciones contextuales robustas

Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *Proceedings of NAACL 2019, volume 1 (long and short papers)*. 2019.

BERT: Bidirectional Encoder Representations from Transformers



Next Sentence Prediction

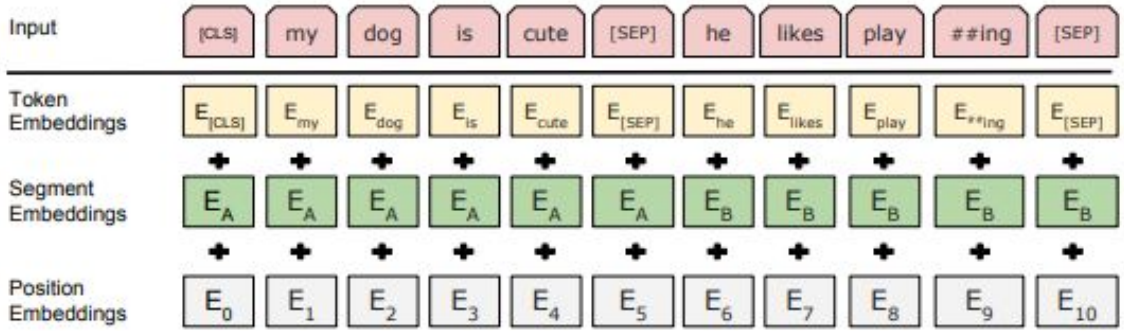
- Dadas dos oraciones, predecir si son contiguas
- Token especial
  - [SEP] para separar oraciones
  - [CLS] para codificar la predicción

- Estudios posteriores argumentan que esta tarea no es necesaria (*)

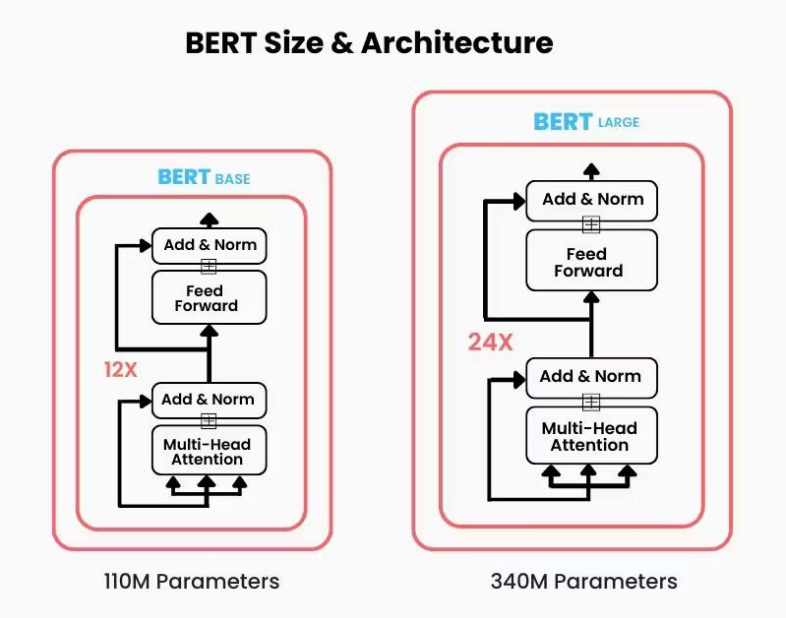
(*) Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692* (2019).

Tokenización y Embeddings en BERT

- Representación de cada unidad sub-palabra contiene
  - Token Emb.: representacion ***semantica***
  - Segment Emb. : identifica a cuál oración pertenece
    - Oracion par: 0 , oracion impar: 1
  - Position Embedding: entrenadas, maxima position=512

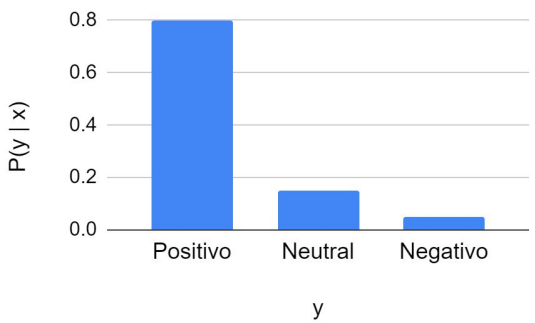


BERT: mas detalles

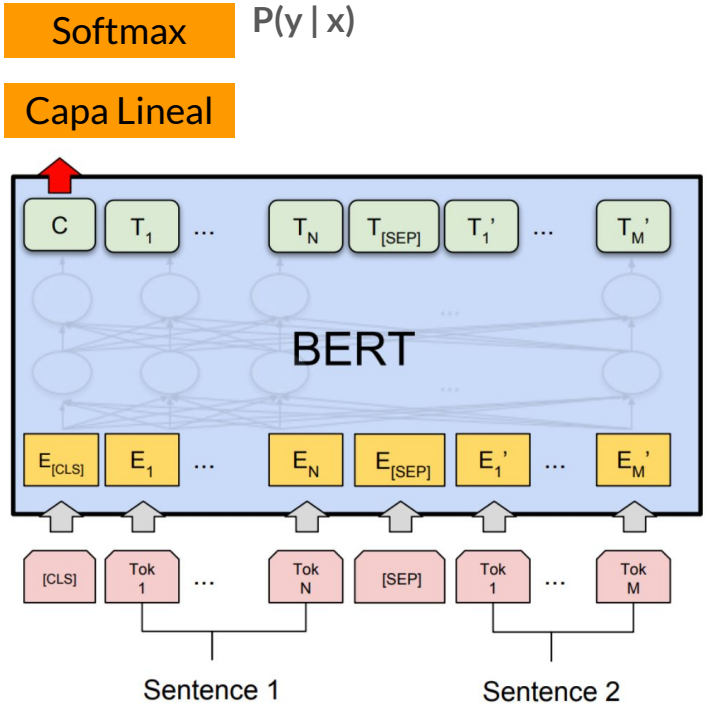


- **BERT-base**
  12 capas, estados de 768 dimensiones,
  12 attention heads, 110M parametros
- **BERT-large**
  24 capas, estados de 1024 dimensiones,
  12 attention heads, 340M parametros
- **Data de pre-entrenamiento**
  - BookCorpus (800M palabras)
  - English Wikipedia (2.5B palabras)

Clasificación de Texto con BERT



Tarea ejemplo:
Sentiment Analysis
(análisis de opinión)



BERT: performance en tareas NLP

- BERT llego a ser increiblemente popular, super versatil
- BERT + finetuning alcanzó performance SOTA en variedad de tareas NLP

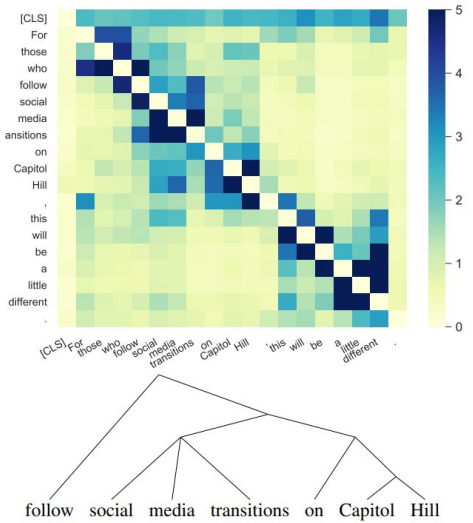
System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

MNLI / QNLI / RTE: natural language inference
SST-2: sentiment analysis
CoLA: gramaticalidad

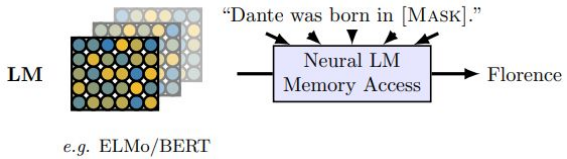
STS-B: similitud semantica
MRPC: parafrasis

BERT-ology

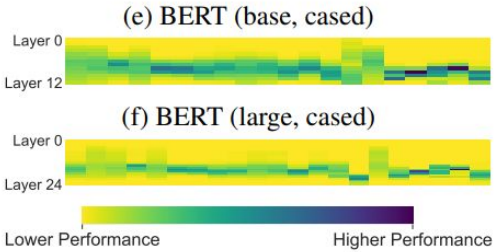
Información sintáctica



Datos Enciclopedicos



Cual capa influencia más durante finetuning? (transfer learning)



Rogers, Anna, Olga Kovaleva, and Anna Rumshisky. "A primer in BERTology: What we know about how BERT works." *Transactions of the association for computational linguistics* 8 (2021): 842-866.

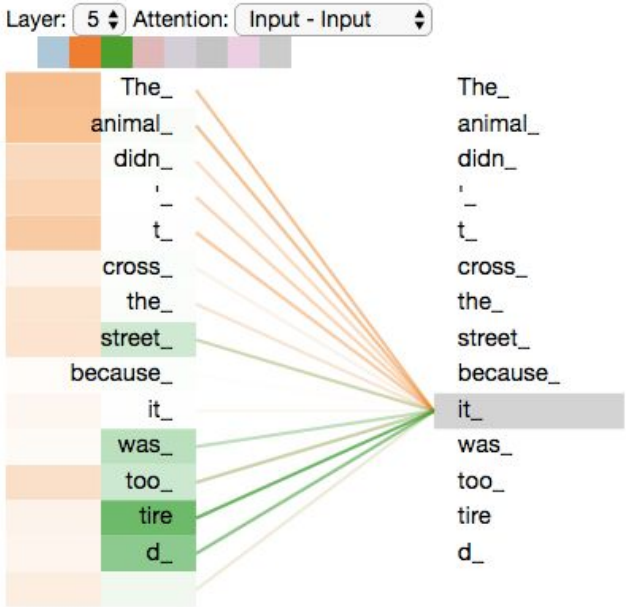
BERT: Recursos de visualizacion y analisis

https://github.com/jessevig/bertviz

https://colab.research.google.com/drive/1hXIQ77A4TYS4y3UthWF-Ci7V7vVUoxmQ?usp=sharing

https://huggingface.co/learn/nlp-course/en/chapter6/5?fw=pt

A explorar!



BERT: variantes

- RoBERTa (https://arxiv.org/abs/1907.11692)
  - BERT + mas data + ajuste de hyper-parametros (batch size, learning rate, warm-up)

- ELECTRA (https://arxiv.org/abs/2003.10555)
  - Generador (MLM) + discriminator (ELECTRA), entrenados para detectar si token ha sido reemplazado

...

Mas informacion:

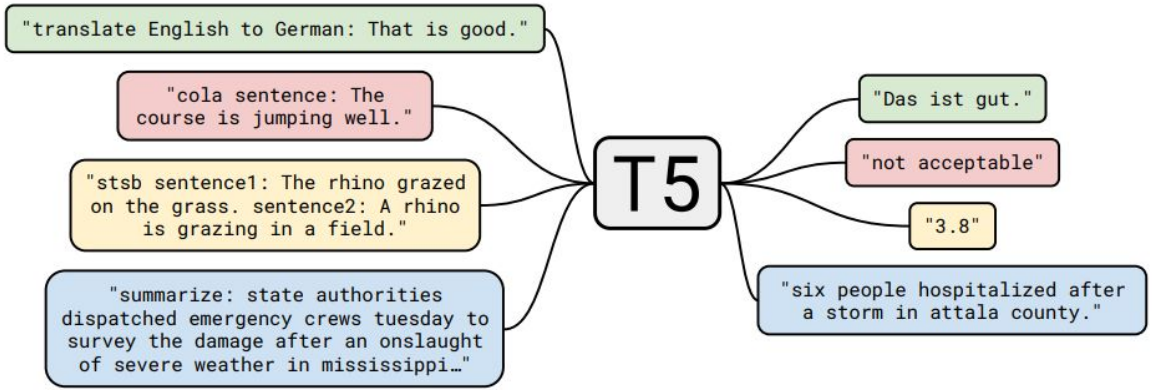
https://txsun1997.github.io/papers/pretrain-survey.pdf

Finetuning BERT: demo

https://colab.research.google.com/drive/1t9oqsp3AB36_BVG1FTc5_CFeps0p58MF?usp=drive_link

Pretraining Encoder-Decoders

Pre-entrenando Encoder-Decoders: T5



Text **T**o **T**ext **T**ransfer **T**ransformer = T5

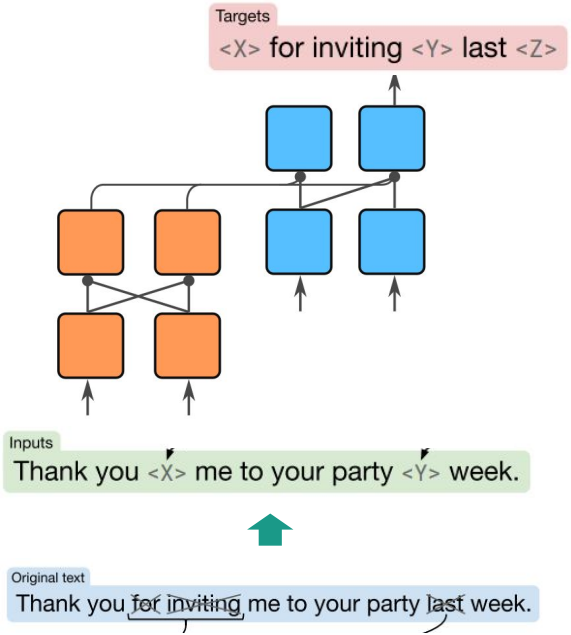
Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *Journal of machine learning research* 21.140 (2020): 1-67.

T5: Pre-entrenamiento

Corrupción de Span o Denoising (Raffel et al.2018)

- Reemplaza bloques de texto de diferente longitud en el input con placeholders
- Genera los bloques que fueron removidos
- Desde el lado del decoder, la tarea es *similar* a Modelado de Lenguaje

- Dataset: texto minado del internet
  - ***C**olossal **C**leaned **C**rawled **C**orpus (C4)*
  - 156B tokens, 745GB



Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *Journal of machine learning research* 21.140 (2020): 1-67.

T5: Finetuning

Texto de entrada (articulo de noticias)

No solo de pan vive el hombre, ni los creadores de contenido gastronómico. Quienes piensan que es sencillo …

Texto a predecir (resumen)

summarize: Influencers gastronómicos: ¿el 'hobbie' de publicar videos de comida puede llegar a ser un oficio rentable

- Se agrega un prefijo especial al inicio del texto a predecir
  - Un prefijo por tarea: *summarize*, *translate*, …
  - Combina los datasets de tantas tareas como sea posible
- Finetune en **X** tareas, evalúa en las mismas **X** tareas

https://elcomercio.pe/somos/historias/influencers-gastronomicos-el-hobbie-de-publicar-videos-de-comida-puede-llegar-a-ser-un-oficio-rentable-trabajos-ofertas-laborales-profesional-futuro-noticia/

T5: Finetuning

Architecture	Objective	Params	Cost	GLUE	CNNDM	SQuAD	SGLUE	EnDe	EnFr	EnRo
★ Encoder-decoder	Denoising	$2P$	M	**83.28**	**19.24**	**80.88**	**71.36**	**26.98**	**39.82**	**27.65**
Enc-dec, shared	Denoising	P	M	82.81	18.78	**80.63**	**70.73**	26.72	39.03	**27.46**
Enc-dec, 6 layers	Denoising	P	$M/2$	80.88	18.97	77.59	68.42	26.38	38.40	26.95
Language model	Denoising	P	M	74.70	17.93	61.14	55.02	25.09	35.28	25.86
Prefix LM	Denoising	P	M	81.82	18.61	78.94	68.11	26.43	37.98	27.39
Encoder-decoder	LM	$2P$	M	79.56	18.59	76.02	64.29	26.27	39.17	26.86
Enc-dec, shared	LM	P	M	79.60	18.13	76.35	63.50	26.62	39.17	27.05
Enc-dec, 6 layers	LM	P	$M/2$	78.67	18.26	75.32	64.06	26.13	38.42	26.89
Language model	LM	P	M	73.78	17.54	53.81	56.51	25.23	34.31	25.38
Prefix LM	LM	P	M	79.68	17.84	76.87	64.86	26.28	37.51	26.76

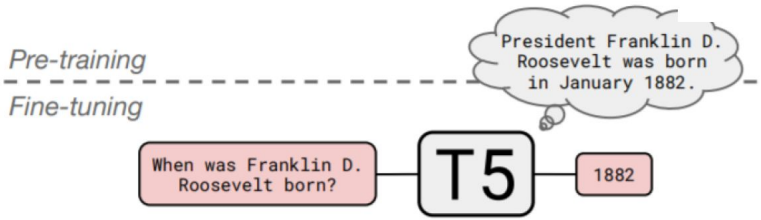
GLUE, SGLUE: Comprension, Parafrasis, Gramatica
CNN-DM: Summarization

EnDe, EnFrm EnRo: Machine Translation
SQuAD: Question Answering

Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *Journal of machine learning research* 21.140 (2020): 1-67.

T5: Finetuning

- T5 es especialmente efectivo en responder preguntas de dominio abierto ("*open-domain* QA")
- Finetuning enseña al modelo a obtener información desde dentro de sus parámetros



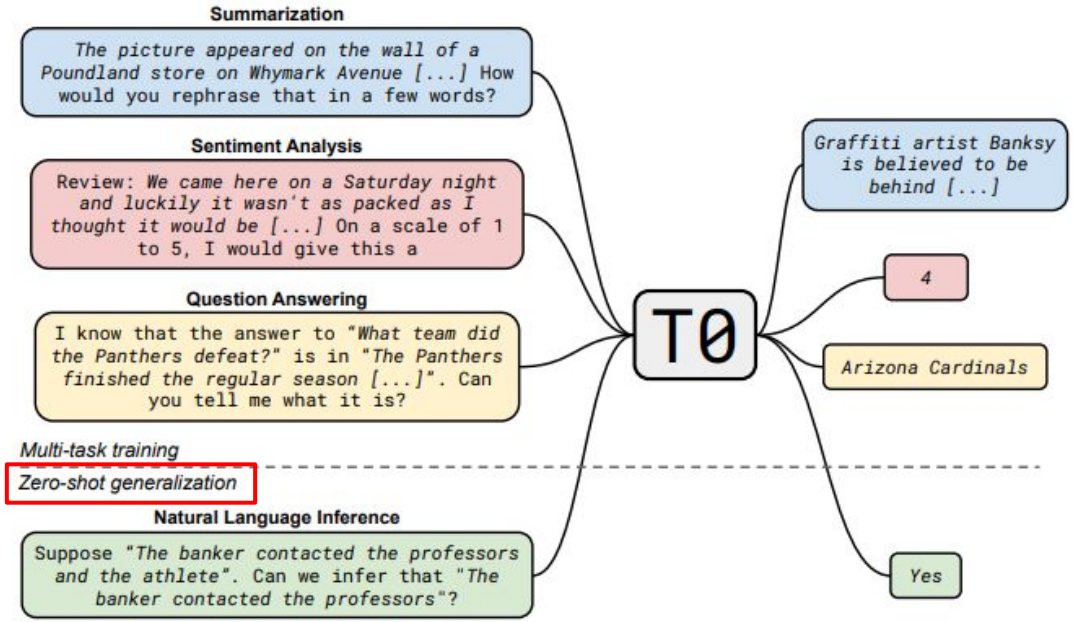
NQ: Natural Questions
WQ: WebQuestions
TQA: TriviaQA

	NQ	WQ	TQA	
			dev	test
Chen et al. (2017)	–	20.7	–	–
Lee et al. (2019)	33.3	36.4	47.1	–
Min et al. (2019a)	28.1	–	50.9	–
Min et al. (2019b)	31.8	31.6	55.4	–
Asai et al. (2019)	32.6	–	–	–
Ling et al. (2020)	–	–	35.7	–
Guu et al. (2020)	40.4	40.7	–	–
Févry et al. (2020)	–	–	43.2	53.4
Karpukhin et al. (2020)	**41.5**	42.4	**57.9**	–
T5-Base	25.9	27.9	23.8	29.1
T5-Large	28.5	30.6	28.7	35.9
T5-3B	30.4	33.6	35.1	43.4
T5-11B	32.6	37.2	42.3	50.1
T5-11B + SSM	34.8	40.8	51.0	60.5
T5.1.1-Base	25.7	28.2	24.2	30.6
T5.1.1-Large	27.3	29.5	28.5	37.2
T5.1.1-XL	29.5	32.4	36.0	45.1
T5.1.1-XXL	32.8	35.6	42.9	52.5
T5.1.1-XXL + SSM	35.2	**42.8**	51.9	**61.6**

https://huggingface.co/google/t5-small-ssm-nq

Roberts, Adam, Colin Raffel, and Noam Shazeer. "How Much Knowledge Can You Pack Into the Parameters of a Language Model?." *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020.

T0: Pretraining + Funetining



Sanh, Victor, et al. "Multitask Prompted Training Enables Zero-Shot Task Generalization." *International Conference on Learning Representations*. 2022.

T0: Pretraining + Funetining

- Basado en T5, capaz de realizar ***zero-shot***
  - Performance comparable a *GPT-3*
- Pretraining
  - Igual que T5 (C4)
- Finetuning
  - Entrena en X tareas, evalua en Y tareas
  - X != Y -> ***zero-shot evaluation***
- Atención especial a ***contaminación de data***
  - NLP datasets = datos del internet anotados con info. linguistica
  - Alta chance de que parte del dataset haya sido memorizada durante pretraining
  - Solucion: decontamina / filtra instancias en datasets con alto overlap

Sanh, Victor, et al. "Multitask Prompted Training Enables Zero-Shot Task Generalization." *International Conference on Learning Representations*. 2022.

T0: Pretraining + Funetining

- Adopta ***Prompting*** en vez de *task prefixing*

Texto de entrada

{Descripción de la tarea}

{Texto de entrada de la instancia}

{Identificador de respuesta}

Texto a predecir

<answer text>

Sanh, Victor, et al. "Multitask Prompted Training Enables Zero-Shot Task Generalization." *International Conference on Learning Representations*. 2022.

T0: Pretraining + Funetining

- Adopta ***Prompting*** en vez de *task prefixing*

Texto de entrada

Read the context and answer the question by selecting one of the provided choices

Question: …..
Context: ….
Choices: (A) …. (B) …. (C) … (D) ….

Answer:

Texto a predecir

<answer text>

Más sobre prompting en al final de esta unidad

Sanh, Victor, et al. "Multitask Prompted Training Enables Zero-Shot Task Generalization." *International Conference on Learning Representations*. 2022.

Pretraining Decoders

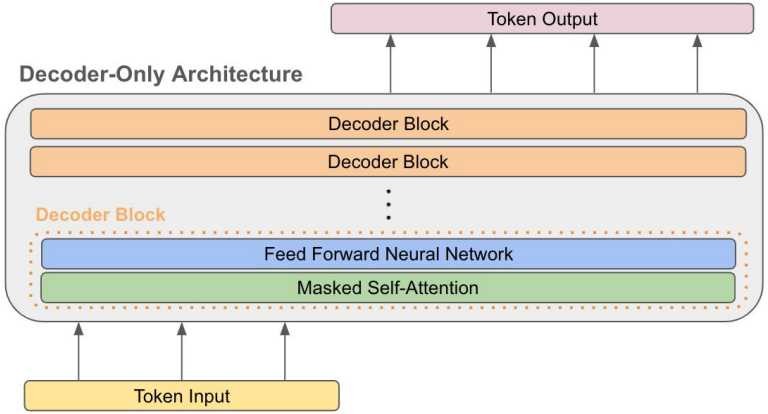
Pre-entrenando Decoders

- Tarea de pre-entrenamiento

  Next Token Prediction (LM)

  $P(w_i \mid w1 \ldots w_{i-1})$

- Útil para tareas que involucran **generar texto**
  - Dialogo: contexto= dialogo previo
  - Resumen: contexto=documento



Generative Pretrained Transformer (GPT)

- Arquitectura
  - Transformer decoder con 12 capas, **117M** parametros
  - Estado de 768 dimensiones, capas MLP/FF con 3072 dimensiones
  - *Causal Attention*
- Vocabulario con 40,000 piezas BPE

- Pre-entrenado en Bookcorpus: +7000 libros
  - Texto contiguo largo, crucial para aprender dependencias lingüísticas distantes

Radford, Alec, et al. "Improving language understanding by generative pre-training." **(2018)**.

GPT: Finetuning

- Utiliza tokens especiales para separar campos de entrada / salida
  - **[START]:** al inicio de toda secuencia
  - **[DELIM]:** entre textos si la tarea involucra comparar dos o mas textos (parafrasis, inferencia,etc)
  - **[EXTRACT]:** para tareas de clasificación, se usa su estado como entrada a un clasificador lineal (similar a BERT)

Tarea	Texto entrada
NLI	[START] El home está en el umbral [DELIM] La persona está cerca de la puerta [EXTRACT]
Sentiment Analysis	[START] La comida estuvo deliciosa [EXTRACT]

Radford, Alec, et al. "Improving language understanding by generative pre-training." **(2018)**.

GPT: Finetuning

Method	Classification		Semantic Similarity			GLUE
	CoLA (mc)	SST2 (acc)	MRPC (F1)	STSB (pc)	QQP (F1)	
Sparse byte mLSTM [16]	-	**93.2**	-	-	-	-
TF-KLD [23]	-	-	**86.0**	-	-	-
ECNU (mixed ensemble) [60]	-	-	-	81.0	-	-
Single-task BiLSTM + ELMo + Attn [64]	35.0	90.2	80.2	55.5	66.1	64.8
Multi-task BiLSTM + ELMo + Attn [64]	18.9	91.6	83.5	72.8	63.3	68.9
GPT Finetuned Transformer LM (ours)	**45.4**	91.3	82.3	**82.0**	**70.3**	**72.8**

Radford, Alec, et al. "Improving language understanding by generative pre-training." **(2018)**.

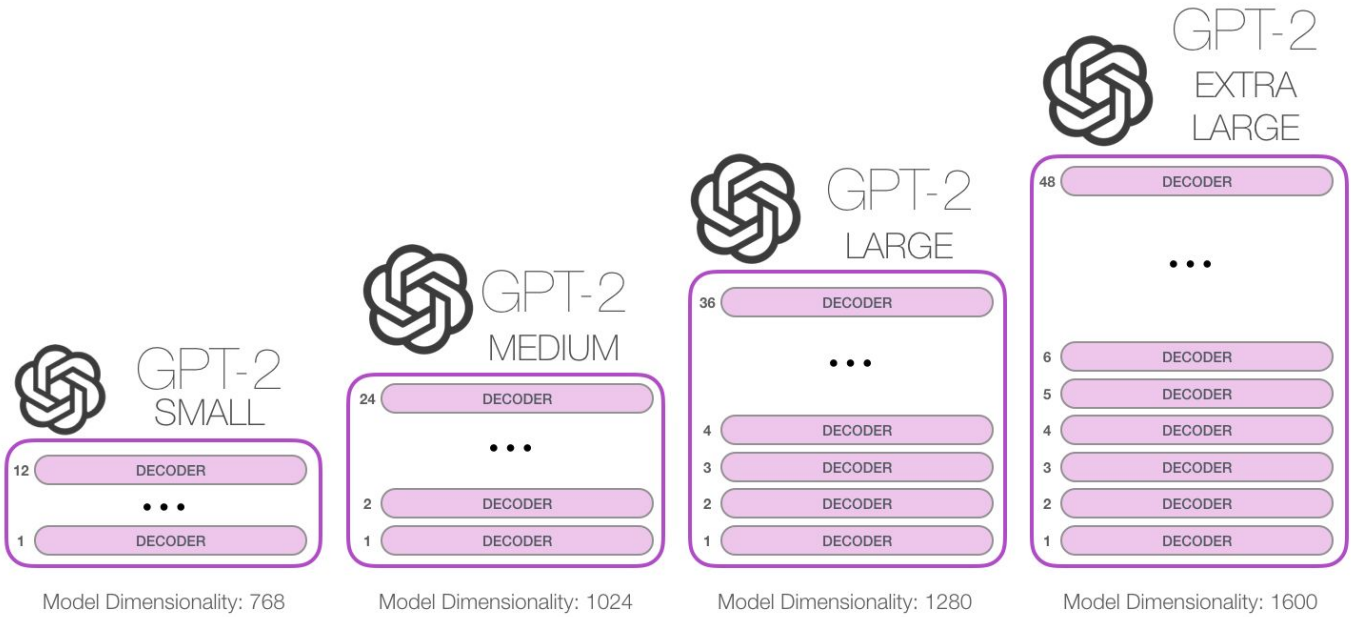
GPT2: mas data + mas parametros

- Arquitectura basada en GPT, escalado a mas parametros
  - Usa embeddings de posicion absoluta
- Pretraining dataset: **WebText (closed-source)**
  - Texto de páginas linkeadas en Reddit (*outbound links*)
  - Filtrada con heurísticas de calidad (e.j. solo posts de Reddit con puntaje medio-alto)
  - Descontaminando / removiendo texto de Wikipedia
  - Resultado: 8M de documentos, ~40GB de texto

https://huggingface.co/openai-community/gpt2

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

GPT2: la familia



Num. parametros	117M	345M	762M	1542M (1.5B)

GPT2

- Dado su tamanho, GPT2 fue el modelo más usado en investigacion de LMs
- Texto generado empieza a ser más coherente, convincente como lenguaje humano

Context (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.
GPT-2: 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.

GPT3: escalando aun mas...

- 175B de parametros
- Data de pre-entrenamiento: 300B de tokens
  - Sampling proporcional a un score de “calidad”

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

- Capaz de resolver tareas con *few-shots*, ejemplos de input/output de la tarea

Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.

Prompting

Zero-shot

In-Context *Learning*

Chain of Thought

Demo

Tecnicas de Prompting

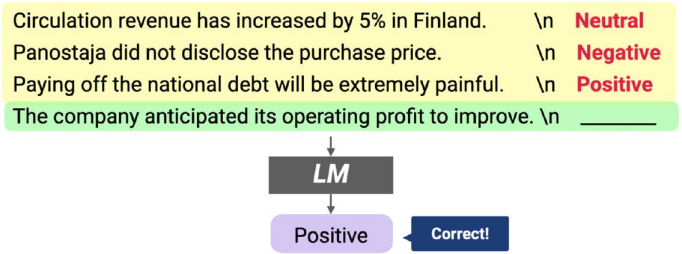
- **Prompting:**
  - Acción de describir la tarea al LLM incluyendo contexto, requerimientos en formato (e.g. json, csv)
- **Tips**
  - Mientras más específica la descripción, mejor
  - Si se dispone de ejemplos, poner unos cuantos

Zero-Shot Prompting

- Zero-Shot == sin ejemplos
- El prompt solo contiene la descripción sin ejemplos
- Útil cuando las descripciones son largas y complejas
  - No hay espacio para ejemplos
  - Recuerde: LLMs tienen un número máximo de tokens que pueden leer

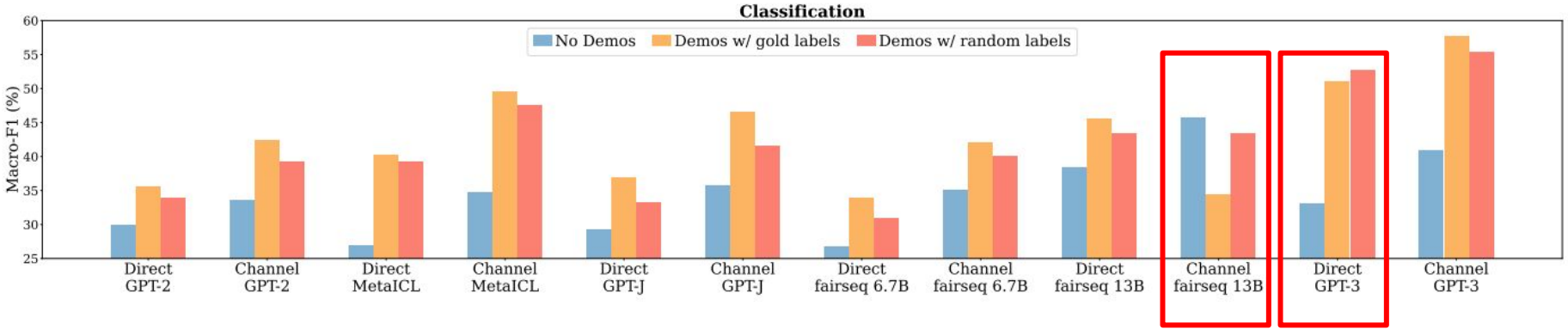
In-Context *Learning* (ICL)

- Tambien llamado: *few-shot* prompting
- El prompt contiene la descripción de la tarea + un número de ejemplos
  - One-shot :: un ejemplo
  - Five-shot :: cinco ejemplos
  - etc
- Útil cuando se desea un formato específico de respuesta
  - Usando palabras fijas (POSITIVO, NEGATIVO)
  - Formato de archivo específico (json, csv)
- **Mejor performance que zero-shot**



In-Context *Learning?*

- No es aprendizaje en el sentido estricto de Machine Learning
  - Pero el *efecto* de procesar ejemplos es similar al efecto de hacer *finetuning*



- Usar gold labels (ejemplos correctos) *sólo es marginalmente mejor* que usar random labels (ejemplos incorrectos)

Work, What Makes In-Context Learning. "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?."

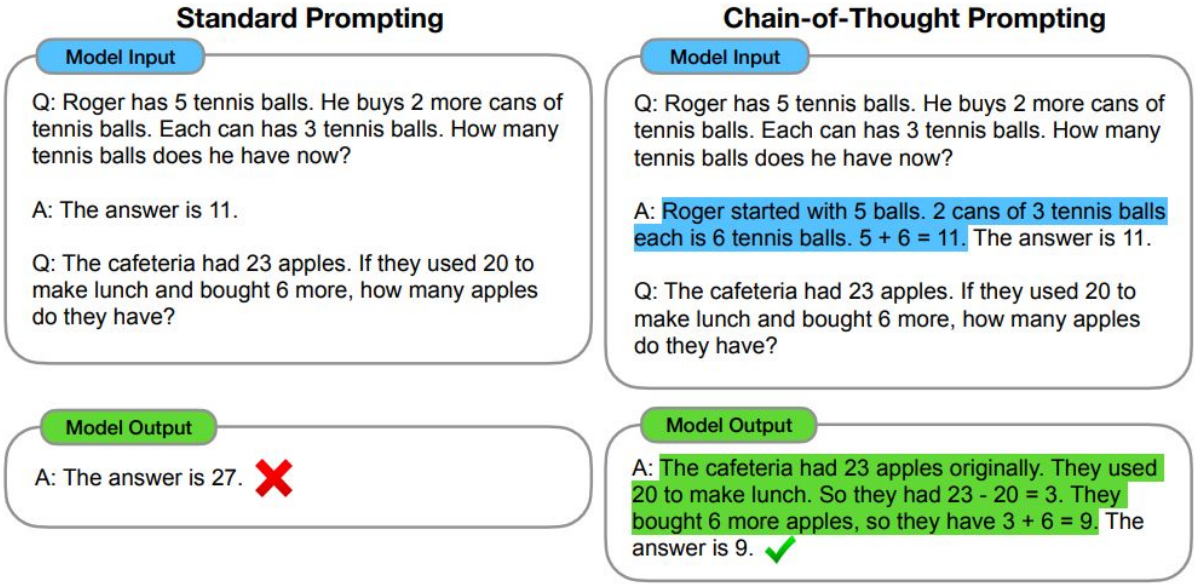
Chain-of-Thought Prompting

- “Tren de pensamiento”
  - Ideado para tareas que requieren de varios pasos de razonamiento
  - Ejemplo: razonamiento matemático

- Consiste pedir al LLM que explique los pasos de razonamiento que llevaron a la respuesta final
  - Más potente con few-shot

Wei, Jason, et al. "Chain-of-thought prompting elicits reasoning in large language models." *Advances in neural information processing systems* 35 (2022): 24824-24837.

Chain-of-Thought Prompting



Wei, Jason, et al. "Chain-of-thought prompting elicits reasoning in large language models." *Advances in neural information processing systems* 35 (2022): 24824-24837.

Lectura adicional: mas alla de CoT prompting

- Exploración de razonamiento paso a paso
  - *Tree* of thoughts: https://arxiv.org/pdf/2305.10601
  - *Graph* of thoughts: https://arxiv.org/pdf/2308.09687

- Darle espacio al LLM a generar pasos intermedios cuando sea necesario
  - Scratchpad generation: https://arxiv.org/pdf/2112.00114

- Formas más elaboradas de exploracion
  - Markov Decision Tree Search: https://arxiv.org/pdf/2406.06592v1

Prompting: Demo

https://colab.research.google.com/drive/10XlXHxNdFudfCXb7bBkXo44YHOFEDUTw?usp=drive_link

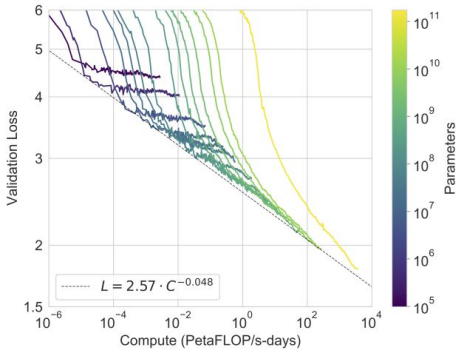
Leyes de Escalamiento de Pretraining

Escalar es efectivo

Mas performance , mejor modelo

- *Escalar* = estrategia efectiva para obtener mejor resultados

Cantidad de data y compute (FLOP) son cruciales!



GPT3

Pre-entrenar con mas data

Orden de billones (GPT3), trillones (Gemma)

Aumentar num. de parametros

GPT (100M) -> GPT3 (175B)
Bloom (+700B)

FLOP = *floating point operation*

Computo

- Medida de la cantidad de operaciones realizadas por un procesador
- Para GPUs, se reporta FLOP
  - Menos computo => mas eficiente

Scaling Laws: Data, Compute, Performance

- Entrenar un LLM desde cero es costoso!
  - Computo (GPUs **$$**), inestabilidad del entrenamiento

- Balance *correcto* entre cantidad de data y computo
  - *Prueba y error* se hace inviable

- ***Existirán reglas que relacionen la cantidad de data, el compute, y el performance?***

investigación al rescate!

Hoffmann, Jordan, et al. "Training compute-optimal large language models." *Proceedings of the 36th International Conference on Neural Information Processing Systems*. 2022.

Scaling Laws: Data, Compute, Performance

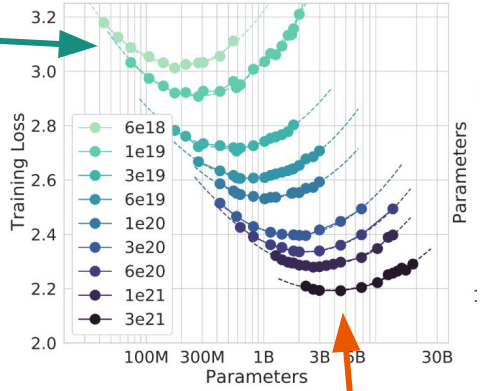
Hoffman et al. (2022)

> “Dado un **presupuesto fijo** de computo (FLOPs), es posible obtener un tamaño de modelo (#parametros) y la cantidad de data (#tokens) de manera que el modelo sea ***óptimamente*** entrenado (training loss)”

Hoffmann, Jordan, et al. "Training compute-optimal large language models." *Proceedings of the 36th International Conference on Neural Information Processing Systems*. 2022.

Scaling Laws: Data, Compute, Performance

Cada curva es un presupuesto en FLOPs distinto



Existe un **#parametros** óptimo con respecto al **loss** (el valle)

Presupuesto total en FLOPs, T,
T= # FLOPs per token-parametro * P * D

donde

- P : # parametros
- D: # tokens en el dataset

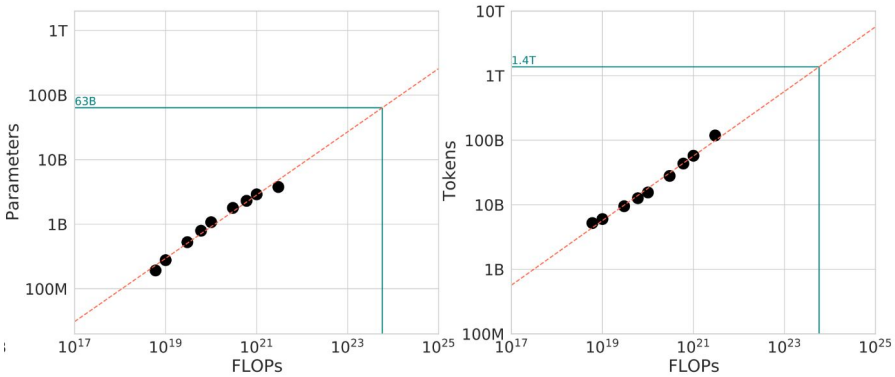
FLOPs per token-parametro: **constante** para determinada arquitectura

Hoffmann, Jordan, et al. "Training compute-optimal large language models." *Proceedings of the 36th International Conference on Neural Information Processing Systems*. 2022.

Como usar el Scaling Laws

- Para cada presupuesto, obten el #param óptimos (P*)
- Para cada presupuesto, obten el #tokens óptimos D*,

$$D^* = T/(\text{FLOPs-t-p} \cdot P^*)$$



- Ajusta un modelo lineal a la data (#params,FLOPs)
- Ajusta un modelo lineal a la data (#tokens,FLOPs)

Para un **número deseado de FLOPs, F**

- Extrapola / interpola hasta F para obtener el #parametros y #tokens óptimos

Hoffmann, Jordan, et al. "Training compute-optimal large language models." *Proceedings of the 36th International Conference on Neural Information Processing Systems*. 2022.

Scaling Laws: Data, Compute, Performance

- Hoffman et al. entrenan un modelo siguiendo este proceso: ***Chinchilla***
- OpenAI entreno GPT-3 optimamente? **No**
  - *under-trained*

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion

Hoffmann, Jordan, et al. "Training compute-optimal large language models." *Proceedings of the 36th International Conference on Neural Information Processing Systems*. 2022.

Metodos de Generacion

Metodos de Generacion

- **Metodos de Busqueda**
  - Greedy Search (Busqueda Avara)
  - Beam Search

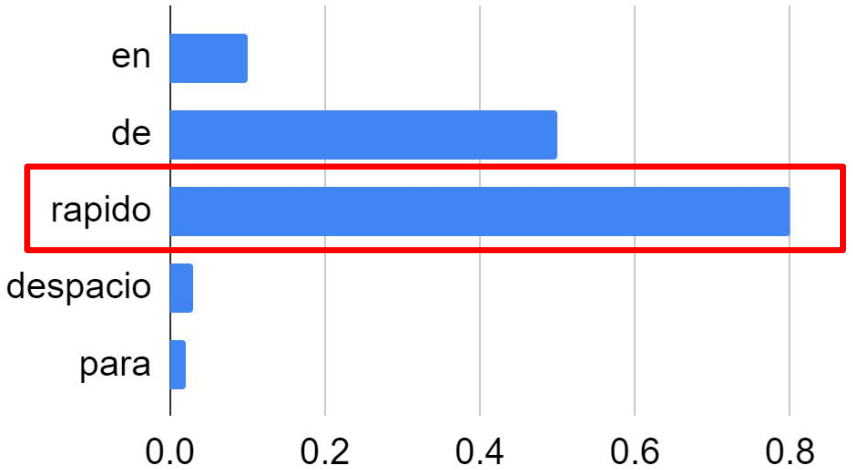
- **Técnicas de muestreo**
  - Muestreo ancestral
  - Seleccion Top-K
  - Muestro Nuclear

Greedy Search (Busqueda Avara)

- Selecciona la palabra con probabilidad **más alta**

“Me gusta correr ___”

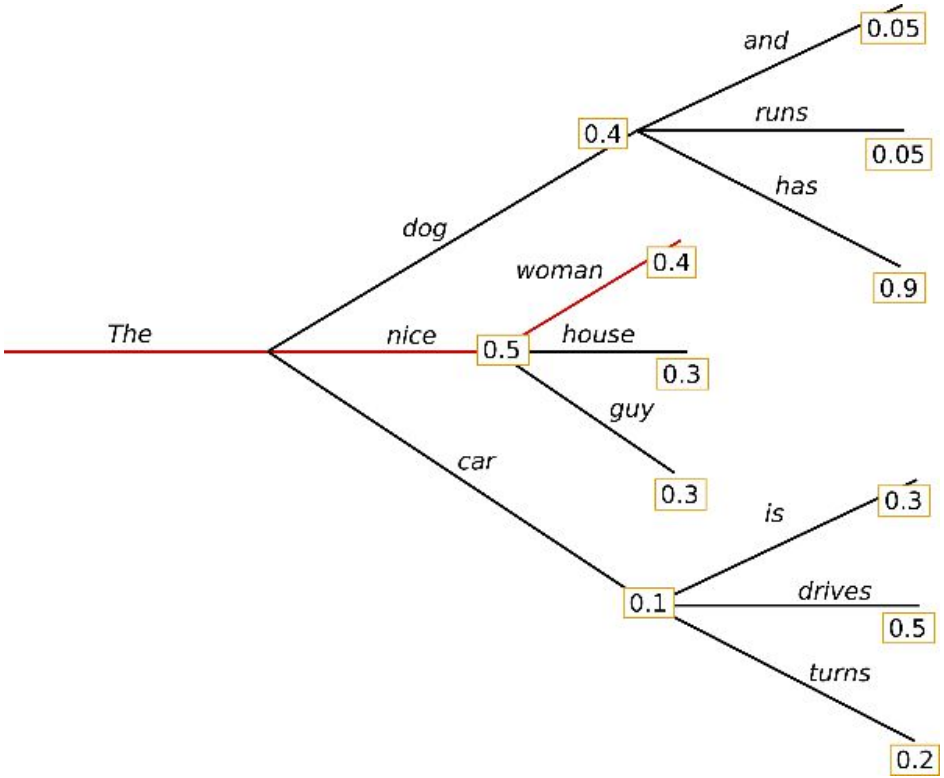
P(* | me,gusta,correr) :



Greedy Search (Busqueda Avara)

- Selecciona la palabra con probabilidad **más alta**
- **Desventaja:**
  - La secuencia final puede no ser la más probable

Greedy Search (Busqueda Avara)



Greedy Search (Busqueda Avara)

- Selecciona la palabra con probabilidad **más alta**
- **Desventaja:**
  - La secuencia final puede no ser la más probable
  - Tiende a producir texto repetitivo

Beam Search (Busqueda en Pila)

- En cada paso, mantiene las **N** secuencias más probables en una pila (stack)

N=2

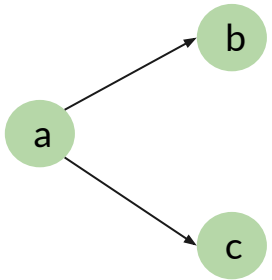
a

Iteracion 0
Beam = {a}

Beam Search (Busqueda en Pila)

- En cada paso, mantiene las **N** secuencias más probables en una pila (stack)

N=2



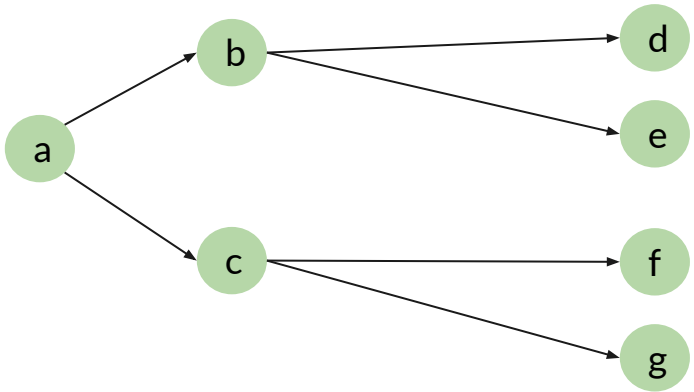
Iteracion 0
Beam = {a}

Iteracion 1
Beam = {ab, ac}

Beam Search (Busqueda en Pila)

- En cada paso, mantiene las **N** secuencias más probables en una pila (stack)

N=2



p(abd) = 0.2

p(abe) = 0.8

p(acf) = 0.6

p(acg) = 0.4

Iteracion 0
Beam = {a}

Iteracion 1
Beam = {ab, ac}

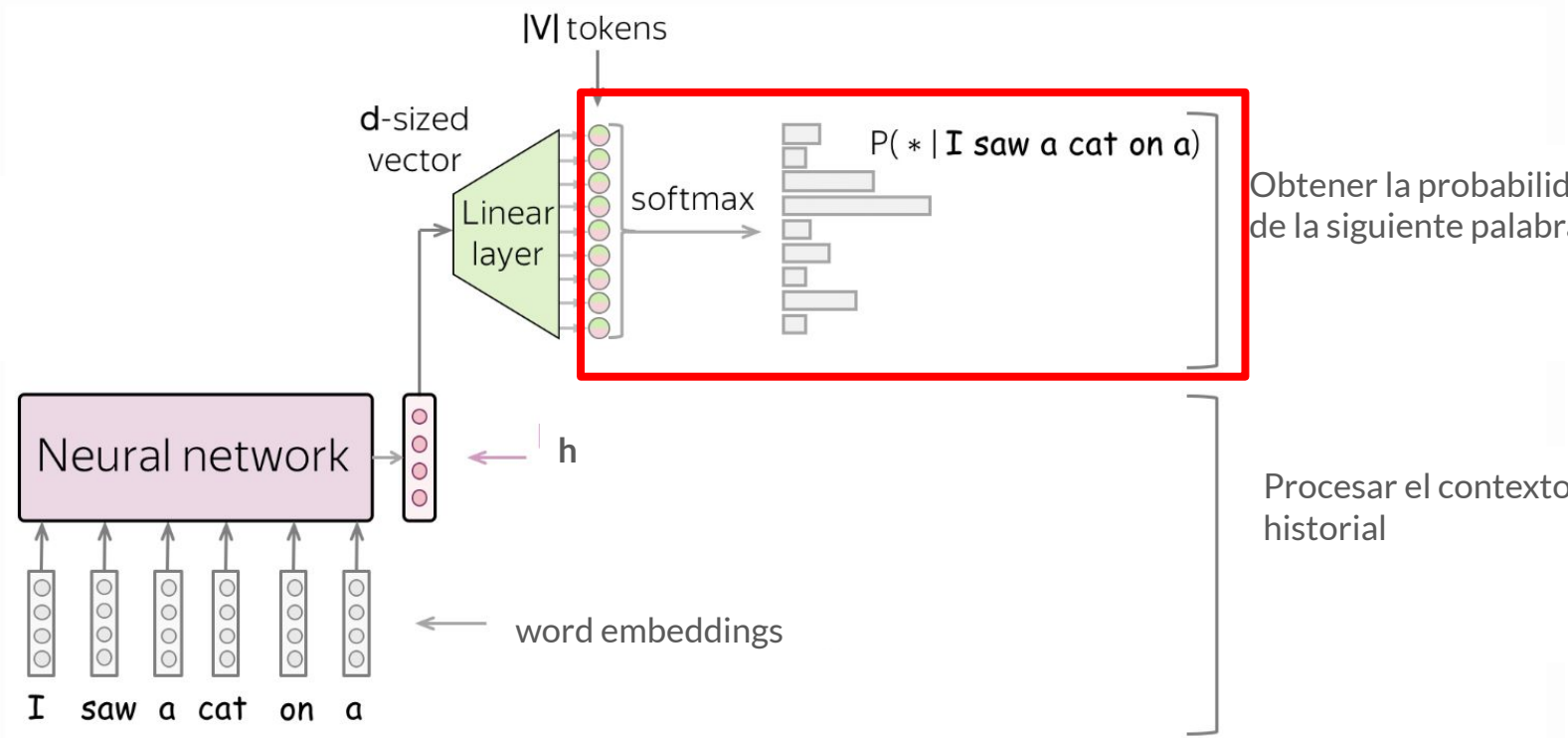
Iteracion 2
Beam = {abe, acf}

Beam Search (Busqueda en Pila)

- En cada paso, mantiene las **N** secuencias más probables en una pila (stack)
- **Ventajas**
  - Secuencias con mayor probabilidad que Greedy Search
  - Texto es más fluido, coherente
- **Desventajas**
  - Las N secuencias pueden no mostrar mucha diversidad léxica
  - Mientras mas alto N, mas costoso computacionalmente.

Generación de Texto: Métodos de Muestreo

Modelos Neurales de Lenguaje



Modelos Neurales de Lenguaje

- La probabilidad de token **a_i** se define como

$$p(a_i|a_{<i}) = softmax(l_i) = \frac{exp(l_i)}{\sum_{j \in V} exp(l_j)}$$

  - **L_i**: score dado a **a_i** por el modelo de lenguaje, aka **logit**

Modelos Neurales de Lenguaje

- La probabilidad de token **a_i** se define como

$$p(a_i|a_{<i}) = softmax(l_i) = \frac{exp(l_i)}{\sum_{j \in V} exp(l_j)}$$

L_i: score dado a **a_i** por el modelo de lenguaje, aka **logit**

- **Desventaja**
  - La distribución puede ser muy desbalanceada

Modelos Neurales de Lenguaje

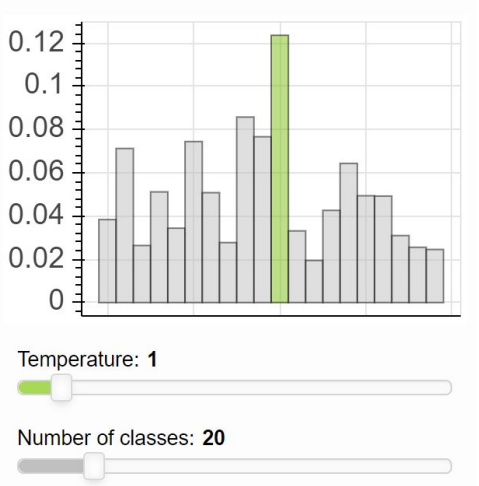
- Desventaja
  - La distribución puede ser muy desbalanceada
  - Solucion: re-escalar la distribucion
- Temperatura tau

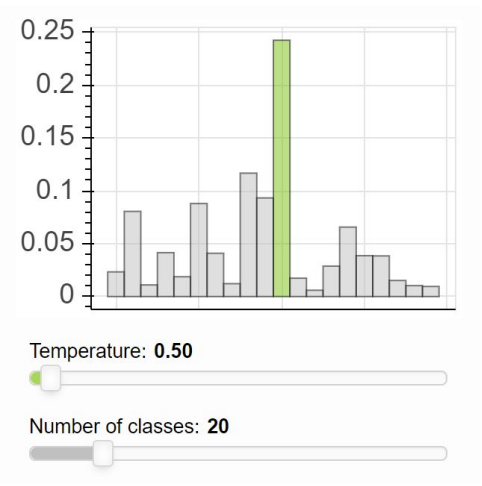
$$p(a_i|a_{<i}) = softmax(l_i/\tau) = \frac{exp(\frac{l_i}{\tau})}{\sum_{j \in V} exp(\frac{l_j}{\tau})}$$

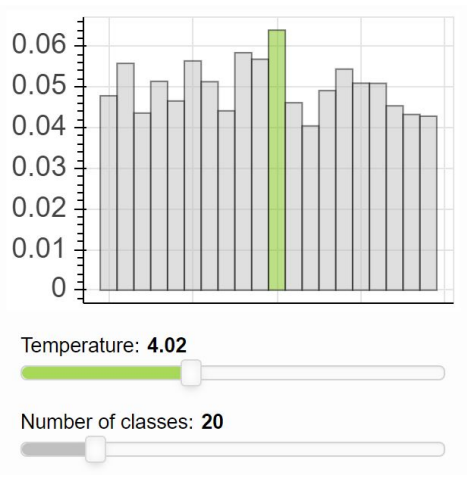
Modelos Neurales de Lenguaje

- Temperatura tau

$$p(a_i|a_{<i}) = softmax(l_i/\tau) = \frac{exp(\frac{l_i}{\tau})}{\sum_{j \in V} exp(\frac{l_j}{\tau})}$$







Ancestral Sampling (Muestro Ancestral)

- En cada paso, muestrea una palabra de nuestro modelo de lenguaje

$$a_i \sim P(*|a_1, \ldots, a_{i-1})$$

- Usualmente usado con re-escalamiento de temperatura
  - Bajo **tau**:
    - menos diversidad léxica
    - mas repetitivo
    - menos fluido
  - Alto **tau**:
    - mas fluido
    - más diversidad léxica
    - Mas chance de generar texto no veraz

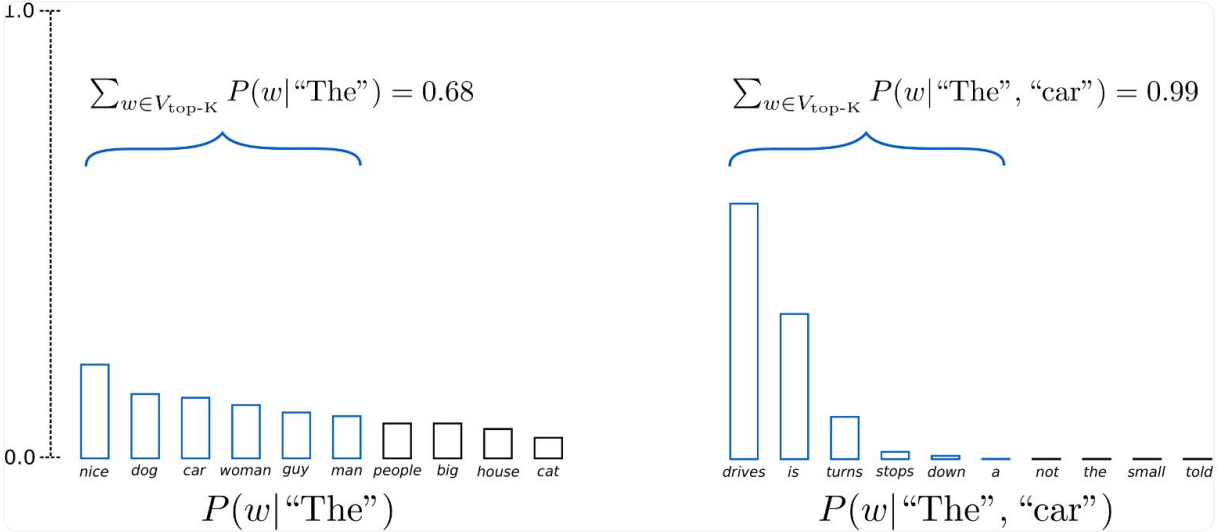
Top-K Sampling

- Antes de muestrear, limita las posibles palabras a las **K** con más probabilidad
  - V -> V_top-k

$$p_{top-k}(a_i|a_{<i}) = \frac{exp(l_i/\tau)}{\sum_{j \in V_{top-k}} exp(l_j/\tau)}$$

Top-K Sampling

- Antes de muestrear, limita las posibles palabras a las **K** con más probabilidad
  - V -> V_top-k



K=6

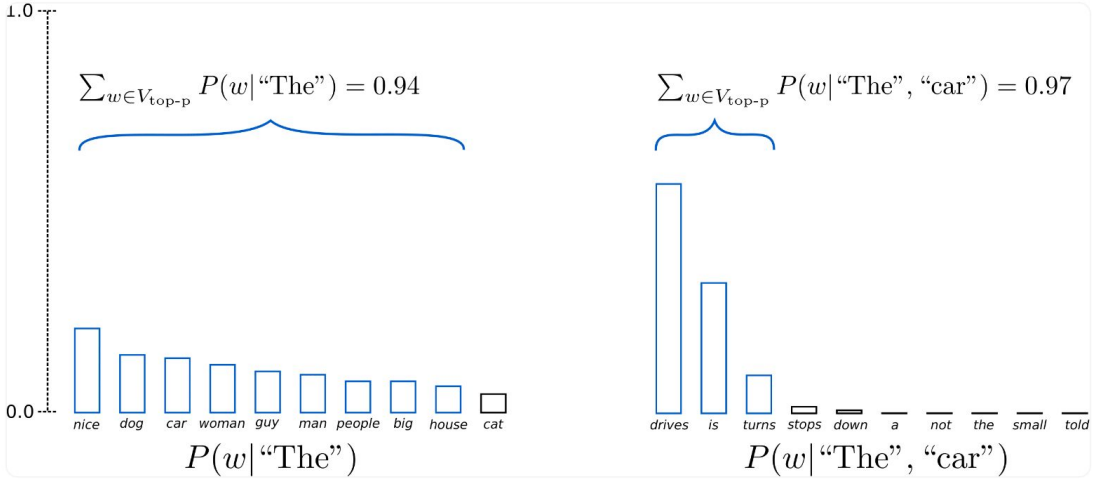
Nucleus Sampling

- Antes de muestrear, limita las posibles palabras al *conjunto que acumule* ***p*** *probabilidad*
  - V -> V_top-p

$$p_{top-p}(a_i|a_{<i}) = \frac{exp(l_i/\tau)}{\sum_{j \in V_{top-p}} exp(l_j/\tau)}$$

Nucleus Sampling

- Antes de muestrear, limita las posibles palabras al *conjunto que acumule* ***p*** *probabilidad*
  - V -> V_top-p



p=0.90

Generando con Texto: Demo

https://colab.research.google.com/drive/1jxlEdjJRXRcKyMWohRznvcrMvS25rCPg?usp=drive_link

Preguntas?