

MoEs, Prompting y evaluación

Docentes:

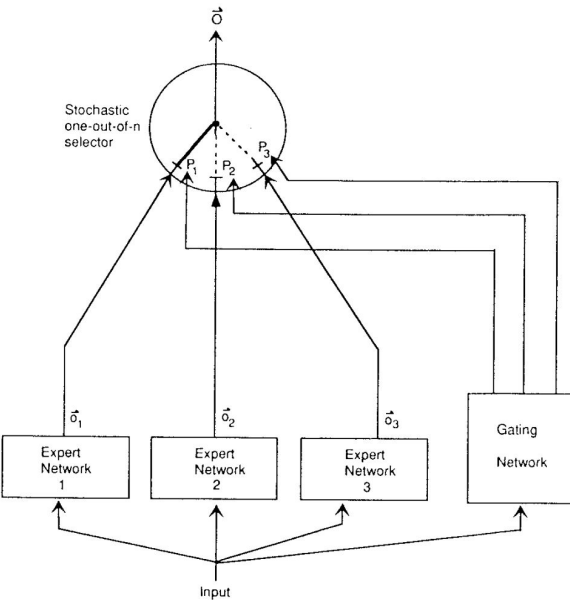
Esp. Ing Abraham Rodriguez - FIUBA

Esp. Ing Ezequiel Guinsburg - FIUBA

Programa de la materia

1. Repaso de Transformers, Arquitectura y Tokenizers.
2. Arquitecturas de LLMs, Transformer Decoder.
3. Ecosistema actual, APIs, costos, HuggingFace y OpenAI.
4. MoEs, técnicas de prompts, evaluación de LLMs.
5. Modelos locales y uso de APIs.
6. RAG, vector DBs, chatbots y práctica.
7. Agentes, fine-tuning y práctica.
8. LLMs de Razonamiento. Optimización, Generación multimodal y práctica.

Mixture of Experts (MoE)



MoE es un concepto introducido en el paper “[Adaptive Mixture of Experts](#)” en 1991, consiste en un conjunto de redes neuronales donde cada una aprende a manejar **subconjuntos** específicos de información.

Cada experto recibe el mismo input y produce el mismo número de outputs, pero existe un mecanismo de selección que **controla** qué expertos contribuyen a la predicción final.

La probabilidad de que se elija la salida de un experto específico j es denotada como p_j .

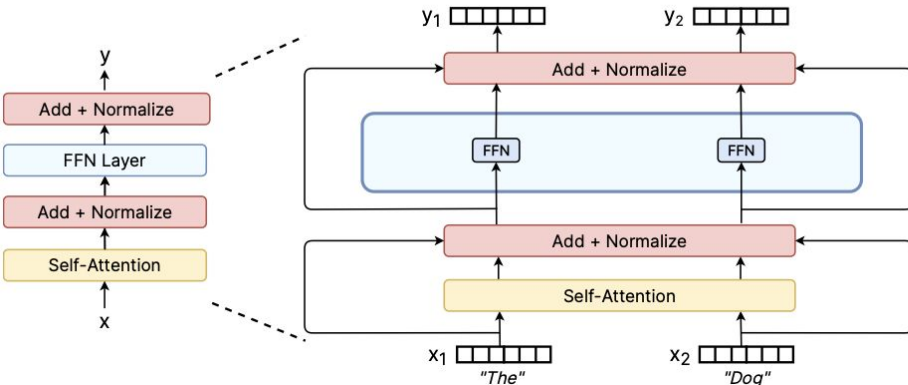
Una red neuronal llamada gating network realiza la selección determinando las contribuciones de cada experto. La gating network aprende las **contribuciones** de cada experto basado en el input.

Esparcidad

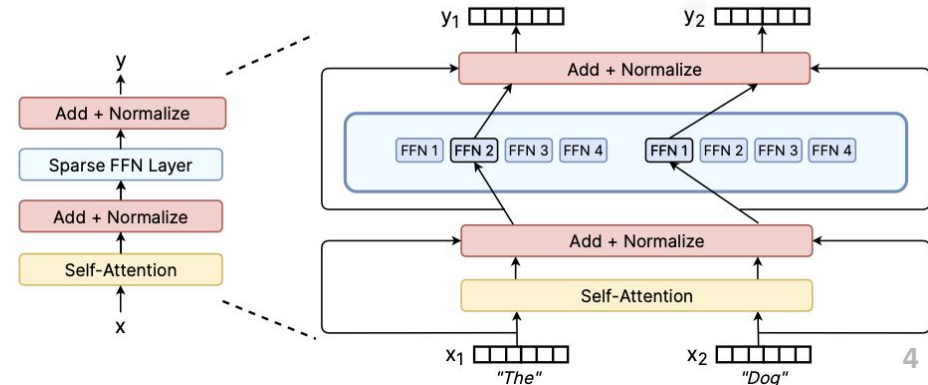
La esparcidad utiliza la idea de **computación condicional**, en un modelo denso todos los parámetros son utilizados para todos los inputs, en un modelo esparzo utiliza parcialmente los parámetros. La gating network G decide que experto E ejecutar el input.

$$y = \sum_{i=1}^n G(x)_i E_i(x) \quad G_{\sigma}(x) = \text{Softmax}(x \cdot W_g)$$

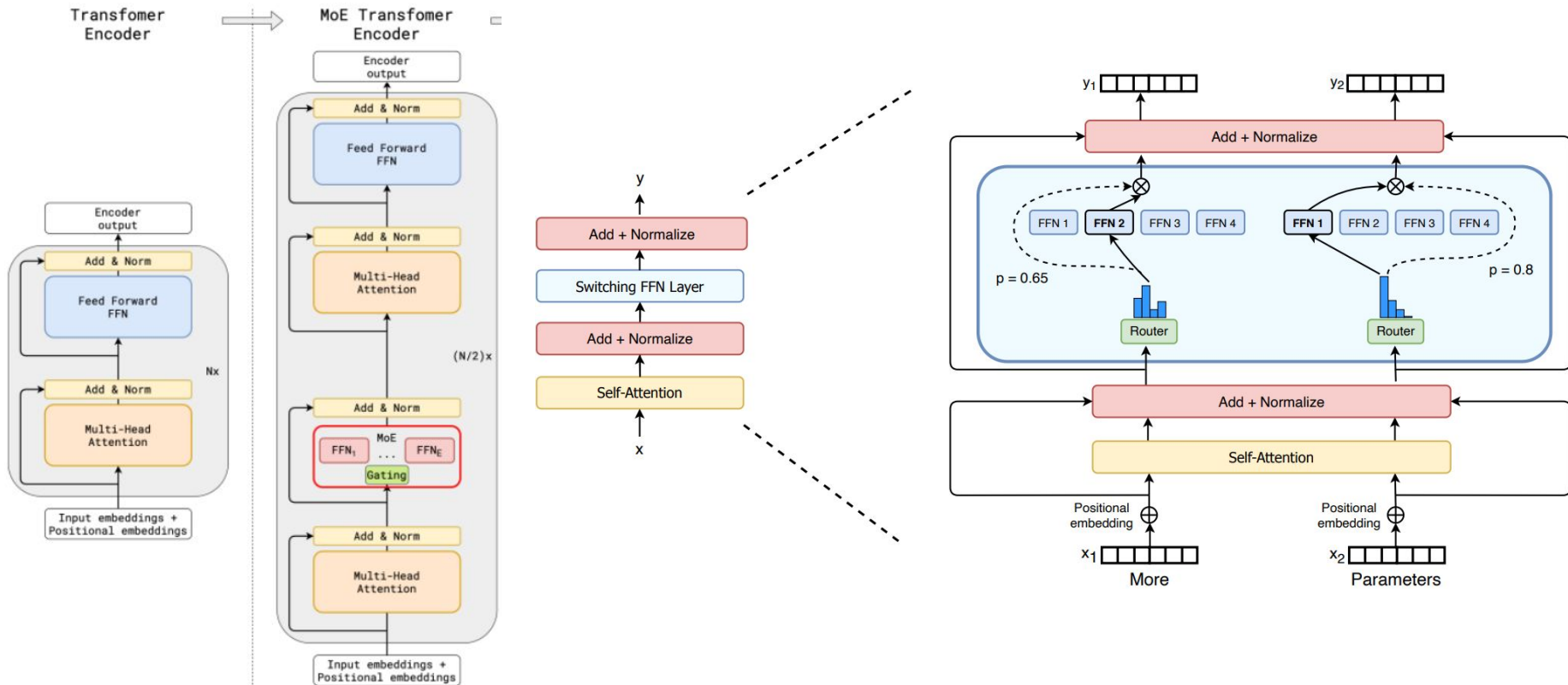
Dense Model



Sparse Model

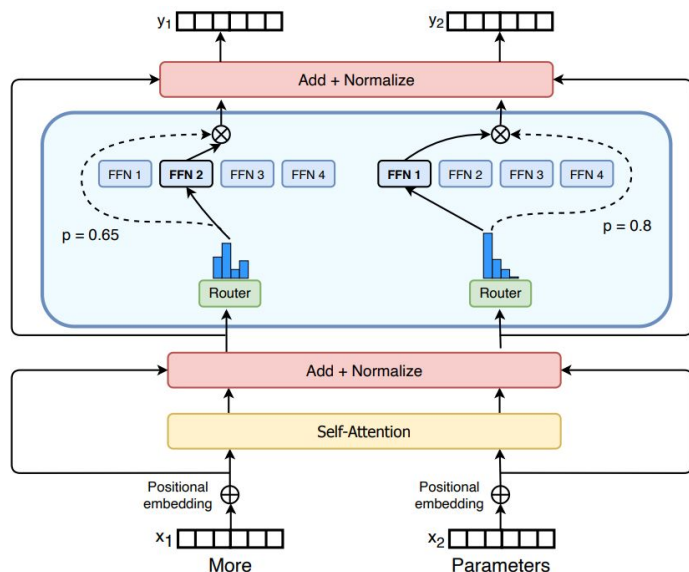


Mixture of Experts (MoE) en LLMs



Mixture of Experts (MoE) en LLMs

Una MoE permite **escalar** los parámetros del modelo sin necesidad de incrementar la demanda computacional, debido al mecanismo de selección dinámico, permitiendo al modelo aloca los recursos de manera condicional.



Mixture of Experts (MoE) en LLMs

MoE Explained

Why new LLMS use MoE

A Survey on Mixture of Experts

El [Paper](#) es un recurso que proporcione un review sobre MoE, técnicas y arquitecturas, altamente **Recomendable leer**.

Sparse Transformers

A REVIEW OF SPARSE EXPERT MODELS IN DEEP
LEARNING **Recomendable leer.**

Mixtral

MistralAI lanzó [Mixtral8x7B](#) (2023), el cual consiste en la misma arquitectura que [Mistral 7B](#), pero con 8 capas FFN Spare MoE.

Mixtral tiene en total 47B de parámetros pero utiliza solamente 13B en inferencia, Mixtral supera a Llama-2-70B y GPT-3.5 en múltiples benchmarks.

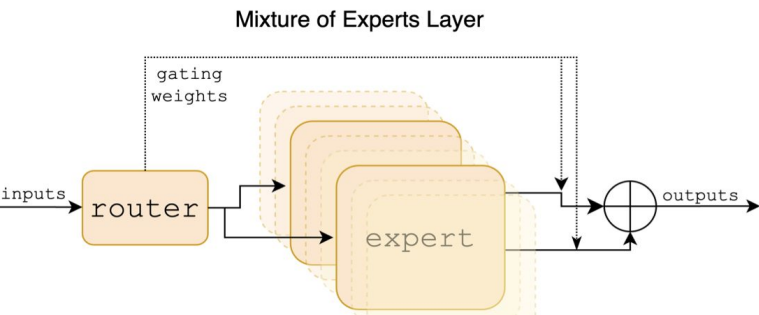
En conjunto se presentó una versión fine-tuned mediante **SFT** de Mixtral-instruct (similar a instructGPT).

[Paper](#)

| | LLaMA 2 70B | GPT - 3.5 | Mixtral 8x7B |
|--|--------------|-------------|--------------|
| MMLU (MCQ in 57 subjects) | 69.9% | 70.0% | 70.6% |
| HellaSwag (10-shot) | 87.1% | 85.5% | 86.7% |
| ARC Challenge (25-shot) | 85.1% | 85.2% | 85.8% |
| WinoGrande (5-shot) | 83.2% | 81.6% | 81.2% |
| MBPP (pass@1) | 49.8% | 52.2% | 60.7% |
| GSM-8K (5-shot) | 53.6% | 57.1% | 58.4% |
| MT Bench (for Instruct Models) | 6.86 | 8.32 | 8.30 |

Mixtral

Mixtral utiliza un router que permite elegir 2 expertos a por token y combinar el output de manera aditiva.



| Parameter | Value |
|---------------|-------|
| dim | 4096 |
| n_layers | 32 |
| head_dim | 128 |
| hidden_dim | 14336 |
| n_heads | 32 |
| n_kv_heads | 8 |
| context_len | 32768 |
| vocab_size | 32000 |
| num_experts | 8 |
| top_k_experts | 2 |

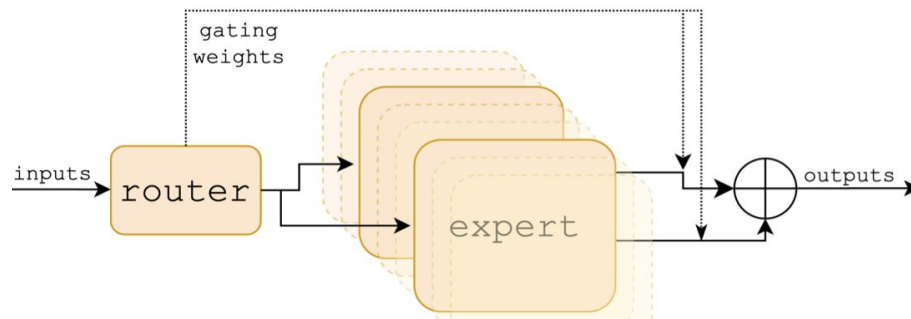
Mixtral Top-K

El router de Mixtral utiliza Top-K logits de una capa lineal, esta técnica fue presentada en el paper “[The Sparsely-Gated MoE Layer](#)”.

$$(\text{TopK}(\ell))_i := \ell_i \quad \text{TopK}(v, k)_i = \begin{cases} v_i & \text{if } v_i \text{ is in the top } k \text{ elements of } v, \\ -\infty & \text{otherwise.} \end{cases}$$

$$G(x) := \text{Softmax}(\text{TopK}(x \cdot W_g))$$

Mixture of Experts Layer



$$y = \sum_{i=0}^{n-1} \text{Softmax}(\text{Top2}(x \cdot W_g))_i \cdot \text{SwiGLU}_i(x)$$

DeepSeek V2

DeepSeekV2 utiliza la idea de expertos compartidos, para reducir la redundancia entre expertos seleccionados,

N_s y N_r denotan el # de shared experts y routed experts.

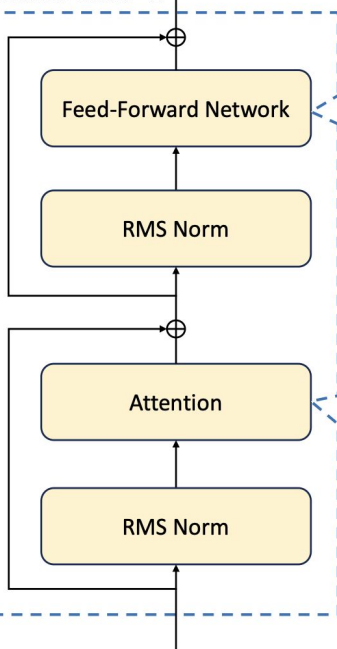
K_r denota el número de experts **activados**.

$$\mathbf{h}'_t = \mathbf{u}_t + \sum_{i=1}^{N_s} \text{FFN}_i^{(s)}(\mathbf{u}_t) + \sum_{i=1}^{N_r} g_{i,t} \text{FFN}_i^{(r)}(\mathbf{u}_t),$$

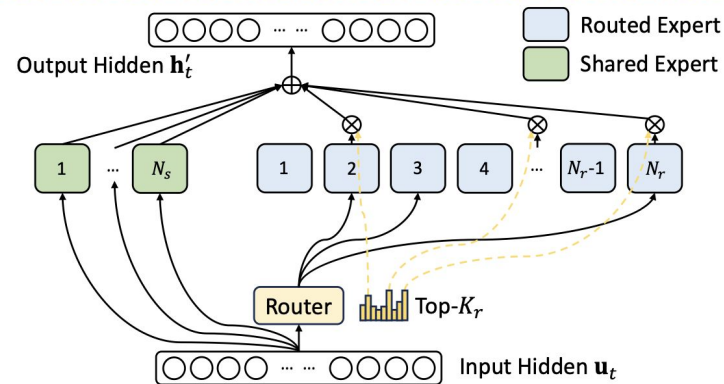
$$g_{i,t} = \begin{cases} s_{i,t}, & s_{i,t} \in \text{Topk}(\{s_{j,t} | 1 \leq j \leq N_r\}, K_r), \\ 0, & \text{otherwise,} \end{cases}$$

$$s_{i,t} = \text{Softmax}_i(\mathbf{u}_t^T \mathbf{e}_i),$$

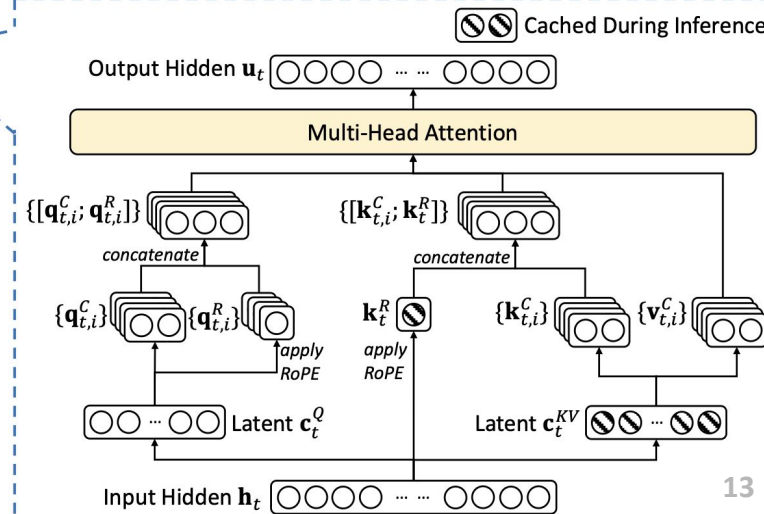
Transformer Block $\times L$



DeepSeekMoE



Multi-Head Latent Attention (MLA)



Grok-1

[Grok-1](#) es una LLM MoE de xAI, la cual no tiene un paper oficial pero en el blog se da a conocer el uso de MoE, métricas, tamaño, etc.

Visual guide to MoE

[Visual guide to MoE](#) (**Recomendable leer**)

MoE Implementaciones

MoE GPT-2

Mistral Transformer layers

Prompting

Chain of Thought

Introducido en el paper “[Chain-of-Thought prompting Elicits Reasoning in Large Language Models](#)”, 2022, introduce el concepto de realizar **pasos intermedios de razonamiento** para habilitar razonamiento complejo de parte de una LLM. Similar al proceso de descomposición que realiza el ahumado sobre problemas de

Standard Prompting

Model Input

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

A: The answer is 11.

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

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

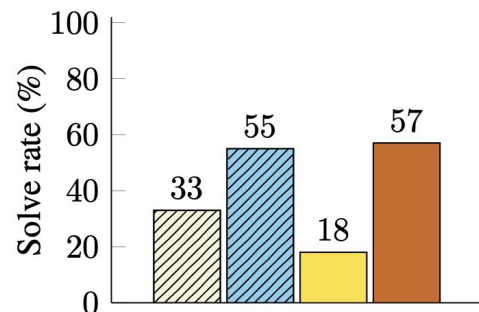
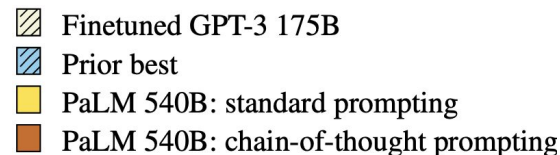
Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

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

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

Model Output

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



Chain of Thought

| | Prompting | GSM8K | SVAMP | ASDiv | AQuA | MAWPS |
|----------------------------------|------------------|-----------------|-------------------|-------------------|-------------------|-------------------|
| Prior best | N/A (finetuning) | 55 ^a | 57.4 ^b | 75.3 ^c | 37.9 ^d | 88.4 ^e |
| UL2 20B | Standard | 4.1 | 10.1 | 16.0 | 20.5 | 16.6 |
| | Chain of thought | 4.4 (+0.3) | 12.5 (+2.4) | 16.9 (+0.9) | 23.6 (+3.1) | 19.1 (+2.5) |
| | + ext. calc | 6.9 | 28.3 | 34.3 | 23.6 | 42.7 |
| LaMDA 137B | Standard | 6.5 | 29.5 | 40.1 | 25.5 | 43.2 |
| | Chain of thought | 14.3 (+7.8) | 37.5 (+8.0) | 46.6 (+6.5) | 20.6 (-4.9) | 57.9 (+14.7) |
| | + ext. calc | 17.8 | 42.1 | 53.4 | 20.6 | 69.3 |
| GPT-3 175B (text-davinci-002) | Standard | 15.6 | 65.7 | 70.3 | 24.8 | 72.7 |
| | Chain of thought | 46.9 (+31.3) | 68.9 (+3.2) | 71.3 (+1.0) | 35.8 (+11.0) | 87.1 (+14.4) |
| | + ext. calc | 49.6 | 70.3 | 71.1 | 35.8 | 87.5 |
| Codex (code-davinci-002) | Standard | 19.7 | 69.9 | 74.0 | 29.5 | 78.7 |
| | Chain of thought | 63.1 (+43.4) | 76.4 (+6.5) | 80.4 (+6.4) | 45.3 (+15.8) | 92.6 (+13.9) |
| | + ext. calc | 65.4 | 77.0 | 80.0 | 45.3 | 93.3 |
| PaLM 540B | Standard | 17.9 | 69.4 | 72.1 | 25.2 | 79.2 |
| | Chain of thought | 56.9 (+39.0) | 79.0 (+9.6) | 73.9 (+1.8) | 35.8 (+10.6) | 93.3 (+14.2) |

En múltiples benchmarks, CoT trae un aumento considerable sobre el razonamiento matemático.

Zero-Shot Chain of Thought

Las LLMs gozan mucho de fama por su capacidad de razonamiento y del razonamiento mediante CoT sin embargo, Zero-shot también puede mejorarse mediante CoT y la frase simple **“Let’s think step by step”**.

Las LLMs gozan mucho de fama y del razonamiento mediante sin embargo, Zero-shot también puede mejorarse mediante CoT y la frase simple “**Let’s think step by step**”.

| | | | | | | | Original GPT-3 (0.3B / 1.3B / 6.7B / 175B) | Instruct GPT-3 (S / M / L / XL-1 / XL-2) | |
|---------------|-------------------|-------------------|------------------|------------------|-----------------------|---------------------|--|--|--------------------------------------|
| | | | | | | | Zero-shot | 2.0 / 1.3 / 1.5 / 3.3 | 3.7 / 3.8 / 4.3 / 8.0 / 17.7 |
| | | | | | | | Few-shot | 5.2 / 5.2 / 4.0 / 8.1 | 3.0 / 2.2 / 4.8 / 14.0 / 33.7 |
| | | | | | | | Zero-shot-CoT | 1.7 / 2.2 / 2.3 / 19.0 | 2.0 / 3.7 / 3.3 / 47.8 / 78.7 |
| | | | | | | | Few-shot-CoT | 4.3 / 1.8 / 6.3 / 44.3 | 2.5 / 2.5 / 3.8 / 36.8 / 93.0 |
| | | | | | | | Zero-Shot | 17.7 | 10.4 |
| | | | | | | | Few-Shot (2 samples) | 33.7 | 15.6 |
| | | | | | | | Few-Shot (8 samples) | 33.8 | 15.6 |
| | | | | | | | Zero-Shot-CoT | 78.7 | 40.7 |
| | | | | | | | Few-Shot-CoT (2 samples) | 84.8 | 41.3 |
| | | | | | | | Few-Shot-CoT (4 samples : First) (*1) | 89.2 | - |
| | | | | | | | Few-Shot-CoT (4 samples : Second) (*1) | 90.5 | - |
| | | | | | | | Few-Shot-CoT (8 samples) | 93.0 | 48.7 |
| | | | | | | | Zero-Plus-Few-Shot-CoT (8 samples) (*2) | 92.8 | 51.5 |
| | | | | | | | Finetuned GPT-3 175B [Wei et al., 2022] | - | 33 |
| | | | | | | | Finetuned GPT-3 175B + verifier [Wei et al., 2022] | - | 55 |
| | | | | | | | PaLM 540B: Zero-Shot | 25.5 | 12.5 |
| | | | | | | | PaLM 540B: Zero-Shot-CoT | 66.1 | 43.0 |
| | | | | | | | PaLM 540B: Zero-Shot-CoT + self consistency | 89.0 | 70.1 |
| | | | | | | | PaLM 540B: Few-Shot [Wei et al., 2022] | - | 17.9 |
| | | | | | | | PaLM 540B: Few-Shot-CoT [Wei et al., 2022] | - | 56.9 |
| | | | | | | | PaLM 540B: Few-Shot-CoT + self consistency [Wang et al., 2022] | - | 74.4 |
| | | | | | | | | | 20 |
| | | | | | | | Arithmetic | | |
| | SingleEq | AddSub | MultiArith | GSM8K | AQUA | SVAMP | | | |
| zero-shot | 74.6/ 78.7 | 72.2/77.0 | 17.7/22.7 | 10.4/12.5 | 22.4/22.4 | 58.8/58.7 | | | |
| zero-shot-cot | 78.0/78.7 | 69.6/74.7 | 78.7/79.3 | 40.7/40.5 | 33.5/31.9 | 62.1/63.7 | | | |
| | | | | | | | Common Sense | | |
| | Common SenseQA | Strategy QA | Date Understand | Shuffled Objects | Last Letter (4 words) | Coin Flip (4 times) | | | |
| zero-shot | 68.8/72.6 | 12.7/ 54.3 | 49.3/33.6 | 31.3/29.7 | 0.2/- | 12.8/53.8 | | | |
| zero-shot-cot | 64.6/64.0 | 54.8/52.3 | 67.5/61.8 | 52.4/52.9 | 57.6/- | 91.4/87.8 | | | |

Zero-Shot Chain of Thought

(a) Few-shot

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

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The answer is 8. ✗

(b) Few-shot-CoT

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

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

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are $16 / 2 = 8$ golf balls. Half of the golf balls are blue. So there are $8 / 2 = 4$ blue golf balls. The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 ✗

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

CoT Zero-Shot vs Few-shot

Hoy en día las LLMs pueden procesar 32k+ tokens, por ejemplo: chatGPT-4, LLama-3 alcanzan los 128k. Esto permite brindar prompts muy grandes. No hace mucho en modelos como Mistral 7B, LLama 2, el context length rondaba en 8k.

Few-shot era poco escalable en la época, ya que los prompts consumen gran cantidad de tokens y recursos computacionales. En este caso Zero-shot es superior. En especial cuando ocupan tareas complejas con context length o recursos finitos.

[Langchain Few-shot examples](#)

[Overcoming Context limit for chatgpt text classification \(2023\)](#)

[The crucial role of context Length in LLM for business applications](#)

Prompt Chaining

Esta técnica es común encontrarla en QA de documentos.

[Prompt chaining](#)

[QA documents with Langchain](#)

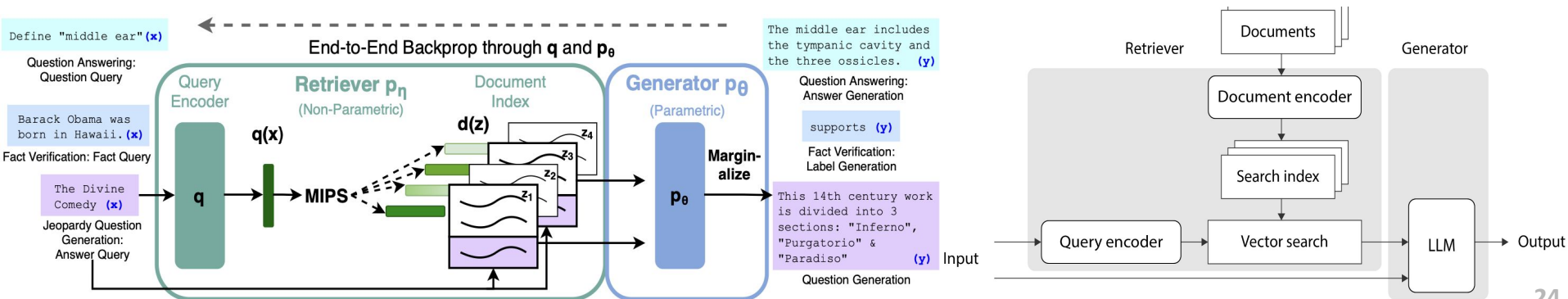
[Document chains Langchain](#)

[Langchain simple chains](#)

[Chain complex prompts for stronger performance](#)

Retrieval Augmented Generation (RAG)

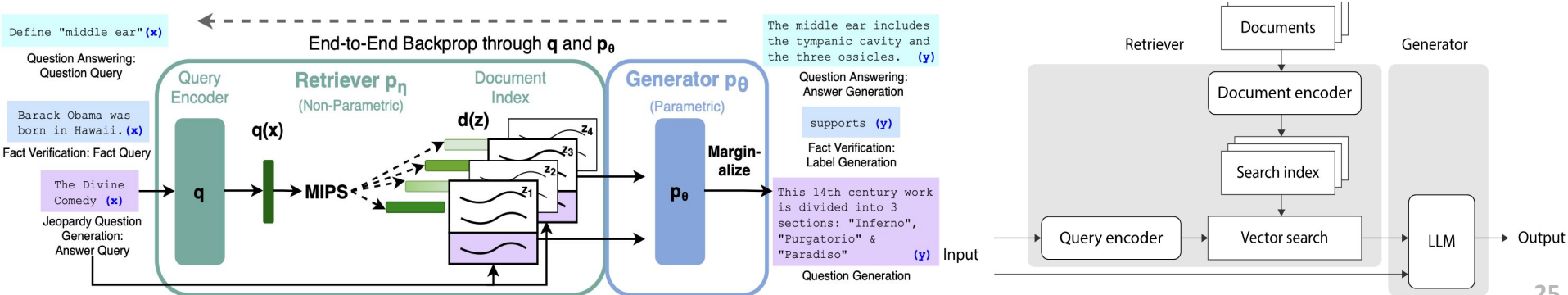
Presentado en el paper “[Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks](#)” en 2020, lo que implica RAG es introducir al modelo conocimiento externo para completar tareas, esto mitiga la alucinación y mejora la confianza de la respuesta.



Retrieval Augmented Generation (RAG)

Presentado en el paper “[Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks](#)” en 2020, RAG implica en **introducir** al modelo contexto nuevo externo para completar tareas, esto mitiga la alucinación y mejora la confianza de la respuesta.

Por ejemplo, documentos privados son desconocidos por la LLM, RAG permite brindar contexto y realizar queries sobre los documentos.



Retrieval Augmented Generation (RAG)

[Databricks RAG](#)

[Pinecone RAG](#)

[Langchain Build a RAG](#)

Evaluación de LLMs

Evaluación de LLMs

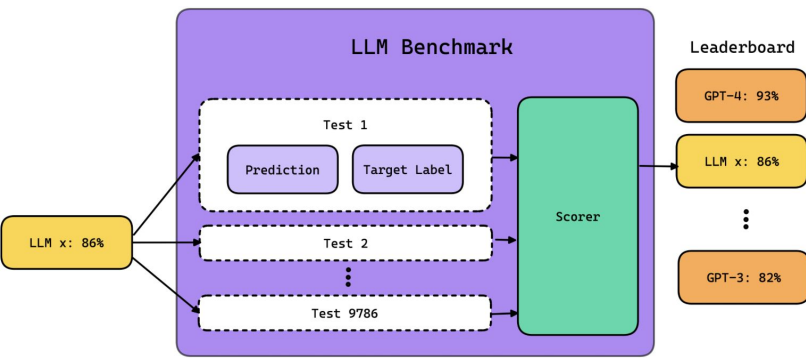
Las LLMs pueden ser evaluadas de manera intrínseca o extrínseca siendo:

| Intrínseca | Extrínseca |
|-----------------------------|--|
| Perplejidad | Benchmark datasets |
| Accuracy | Rendimiento en Zero-shot, Few-shot, etc. |
| Embedding representation | Evaluación humana |

Evaluating LLMs

LLM Benchmarks

Mediante Benchmarks estandarizados es realizable la evaluación medida para **múltiples** tareas.



Razonamiento y sentido común: Capacidad para aplicar lógica y resolver problemas.

Comprensión de lenguaje y preguntas/respuestas: Habilidad para interpretar texto y responder preguntas con precisión.

Codificación: Capacidad para interpretar y generar código.

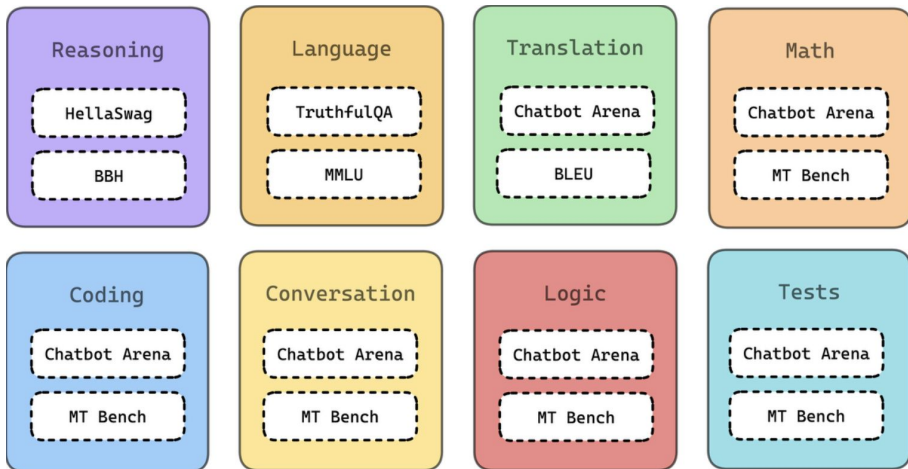
Conversación y chatbots: Capacidad para entablar diálogos y responder de manera coherente y relevante.

Traducción: Habilidad para traducir texto de un idioma a otro con precisión.

Matemáticas: Resolución de problemas matemáticos, desde aritmética básica hasta áreas complejas como cálculo.

Pruebas estandarizadas: Exámenes como el SAT o ACT para evaluar el desempeño del modelo en contextos educativos.

Evaluación de LLMs Benchmarks



| | LLaMA 2 70B | GPT - 3.5 | Mixtral 8x7B |
|--|--------------|-------------|--------------|
| MMLU (MCQ in 57 subjects) | 69.9% | 70.0% | 70.6% |
| HellaSwag (10-shot) | 87.1% | 85.5% | 86.7% |
| ARC Challenge (25-shot) | 85.1% | 85.2% | 85.8% |
| WinoGrande (5-shot) | 83.2% | 81.6% | 81.2% |
| MBPP (pass @ 1) | 49.8% | 52.2% | 60.7% |
| GSM-8K (5-shot) | 53.6% | 57.1% | 58.4% |
| MT Bench (for Instruct Models) | 6.86 | 8.32 | 8.30 |

[LLM benchmarks explained](#)

DeepEval

Es un [framework opensource](#) para evaluar LLMs busca ser fácil de integrar como un “[unit test](#)” similar a Pytest. [Github](#)

[Tutorial](#)

[Documentación DeepEval](#)

Evaluación de LLMs

[Huggingface LLM Eval](#)

[List of eval metrics](#)

[SuperGLUE](#)

Preguntas?