



UNIVERSIDAD DE CHILE

Inteligencia Artificial Generativa

Let's talk about hype stuff

Valentin Barriere // Clemente Henriquez

Universidad de Chile – DCC

Diplomado de Postítulo en Inteligencia Artificial, Primavera 2025

Grandes Modelos de Lenguaje Generativos

Outline : Introducción

Introducción

Habilidades y Aprendizaje en
Contexto

Tokenización y Entrenamiento

Instrucciones y Alineamientos

Razonamientos

Entrenamiento en la práctica

Evaluación de LLMs

LLMs como Agentes

¿Qué es un modelo de lenguaje?

La definición clásica de un modelo de lenguaje (ML) es una **distribución de probabilidad sobre secuencias de tokens**. Supongamos que tenemos un vocabulario \mathcal{V} de un conjunto de tokens. Un modelo de lenguaje p asigna a cada secuencia de tokens $x_1, \dots, x_L \in \mathcal{V}$ una probabilidad (un número entre 0 y 1):

$$p(x_1, \dots, x_L)$$

Intuitivamente, la probabilidad nos indica cuán “buena” es una secuencia de tokens.

Dificultad

La capacidad de asignar probabilidades (significativas) a todas las secuencias **requiere habilidades lingüísticas extraordinarias (pero implícitas) y conocimiento del mundo**.

Modelos de Lenguaje Autoregresivos

Definición General

Un modelo autorregresivo puede usarse para describir ciertos procesos que varían en el tiempo en la naturaleza, la economía o el comportamiento. Especifica que la variable de salida depende **linealmente** de sus propios valores anteriores y de un término estocástico: $X_t = \sum_{i=1}^p \varphi_i X_{t-i} + \epsilon_t$

$$p(x_{1:L}) = p(x_1)p(x_2 | x_1)p(x_3 | x_1, x_2) \cdots p(x_L | x_{1:L-1}) = \prod_{i=1}^L p(x_i | x_{1:i-1}).$$

En particular, $p(x_i | x_{1:i-1})$ es una **distribución de probabilidad condicional** del siguiente token x_i dada la secuencia previa $x_{1:i-1}$.

$$p(\text{the}, \text{mouse}, \text{ate}, \text{the}, \text{cheese}) = p(\text{the})$$

$$p(\text{mouse} | \text{the})$$

$$p(\text{ate} | \text{the}, \text{mouse})$$

$$p(\text{the} | \text{the}, \text{mouse}, \text{ate})$$

$$p(\text{cheese} | \text{the}, \text{mouse}, \text{ate}, \text{the})$$

Temperatura y generación

Para generar una secuencia completa $x_{1:L}$ a partir de un modelo de lenguaje autorregresivo p , muestreamos un token a la vez dado los tokens generados hasta el momento:

$$\text{para } i = 1, \dots, L : \quad x_i \sim p(x_i | x_{1:i-1})^{1/T}$$

donde $T \geq 0$ es un parámetro de **temperatura** que controla cuánta aleatoriedad queremos del modelo de lenguaje:

- $T = 0$: elegir determinísticamente el token más probable x_i en cada posición
- $T = 1$: muestrear “normalmente” del modelo de lenguaje puro
- $T = \infty$: muestrear desde una **distribución uniforme** sobre todo el vocabulario \mathcal{V}

Generación condicional

Más generalmente, podemos realizar generación condicional especificando alguna secuencia prefijo $x_{1:i}$ (llamada **prompt**) y muestreando el resto $x_{i+1:L}$ (llamada la **completación**).

Por ejemplo, generar con $T = 0$ produce:

$$\underbrace{\text{the, mouse, ate}}_{\text{prompt}} \xrightarrow{T=0} \underbrace{\text{the, cheese}}_{\text{completación}}$$

Termodinámica y Entropía

- Una medida del **grado de aleatoriedad** de la energía en un sistema
- Cuanto menor es la entropía, más ordenado y menos aleatorio es

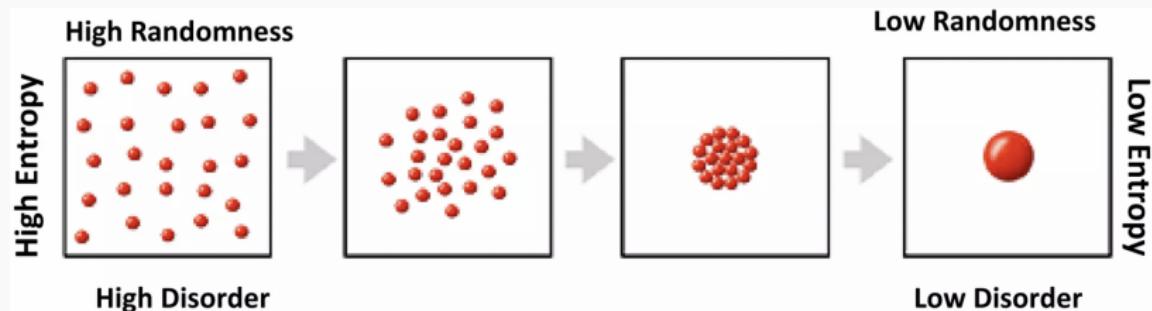


Figure 1: Entropía en termodinámica

Con temperatura y ML

Cuanto mayor es la temperatura de la habitación, más entropía generará. Usando $T = \infty$ obtenemos una distribución uniforme de las probabilidades sobre \mathcal{V} , que es lo más aleatorio posible.

Teoría de la información y Entropía

La entropía de una distribución se define como:

$$H(p) = \sum_x p(x) \log \frac{1}{p(x)}.$$

La entropía mide el número esperado de bits que **cualquier algoritmo** necesita para codificar (comprimir) una muestra $x \sim p$ en una cadena de bits:

the mouse ate the cheese $\Rightarrow 0001110101$.

- Cuanto menor es la entropía, más “estructurada” es la secuencia y más corta es la longitud del código.
- Intuitivamente, $\log \frac{1}{p(x)}$ es la longitud del código usado para representar un elemento x que ocurre con probabilidad $p(x)$.
- Si $p(x) = \frac{1}{8}$, deberíamos asignar $\log_2(8) = 3$ bits (equivalentemente, $\log(8) = 2.08$ nats).

Entropía Cruzada

Entropía cruzada

$$H(p, q) = \sum_x p(x) \log \frac{1}{q(x)},$$

mide el número esperado de bits necesarios para codificar una muestra $x \sim p$ usando el esquema de compresión dado por el modelo q (representando x con un código de longitud $\frac{1}{q(x)}$).

Una propiedad crucial es que la entropía cruzada $H(p, q)$ acota por arriba la entropía $H(p)$:

$$H(p, q) \geq H(p),$$

lo que significa que podemos estimar $H(p, q)$ construyendo un modelo (de lenguaje) q con solo muestras de la verdadera distribución de datos p , mientras que $H(p)$ generalmente es inaccesible si p es el inglés.

Perplejidad

La probabilidad conjunta de una secuencia depende de su longitud y por lo tanto **tiende a cero** conforme la longitud crece, lo que hace difícil su seguimiento. Intuitivamente nos gustaría promediar las probabilidades por token $p(x_i | x_{1:i-1})$.

La perplejidad se define así:

$$\text{perplexity}_p(x_{1:L}) = \exp \left(\frac{1}{L} \sum_{i=1}^L \log \frac{1}{p(x_i | x_{1:i-1})} \right)$$

Reescribiendo la Entropía entre el texto real y las predicciones del ML:

$$H(GT, LM) = \sum_{i=1}^L \sum_{w \in \mathcal{V}} \mathbb{1}_{x_i}(w) \log \frac{1}{p(w | x_{1:i-1})}$$

Obtenemos: $\text{perplexity}_p(x_{1:L}) = \exp \left(\frac{1}{L} H(GT, LM) \right)$ o, $H = \log \text{perplexity}$

Perplejidad

La probabilidad conjunta de una secuencia depende de su longitud y por lo tanto **tiende a cero** conforme la longitud crece, lo que hace difícil su seguimiento. Intuitivamente nos gustaría promediar las probabilidades por token $p(x_i | x_{1:i-1})$.

La perplejidad se define así:

$$\text{perplexity}_p(x_{1:L}) = \exp \left(\frac{1}{L} \sum_{i=1}^L \log \frac{1}{p(x_i | x_{1:i-1})} \right)$$

Reescribiendo la Entropía entre el texto real y las predicciones del ML:

$$H(GT, LM) = \sum_{i=1}^L \sum_{w \in \mathcal{V}} \mathbb{1}_{x_i}(w) \log \frac{1}{p(w | x_{1:i-1})}$$

Obtenemos: $\text{perplexity}_p(x_{1:L}) = \exp \left(\frac{1}{L} H(GT, LM) \right)$ o, $H = \log \text{perplexity}$

Modelos N-grama

En un **modelo n-grama**, la predicción de un token x_i solo **depende de los últimos $n - 1$ caracteres** $x_{i-(n-1):i-1}$ en lugar del historial completo:

$$p(x_i \mid x_{1:i-1}) = p(x_i \mid x_{i-(n-1):i-1}).$$

Por ejemplo, un modelo trigram ($n = 3$) definiría:

$$p(\text{cheese} \mid \text{the, mouse, ate, the}) = p(\text{cheese} \mid \text{ate, the}).$$

Estas probabilidades se calculan en base al número de veces que aparecen varios n-gramas (p. ej., ate the mouse y ate the cheese) en un gran corpus de texto, y se suavizan apropiadamente para evitar sobreajuste.

Problema

Cuando n es demasiado grande, no se puede calcular el n-grama porque depende de ocurrencias y las oraciones largas tienden a ser únicas.

Modelos de lenguaje neuronales

Con la introducción de redes neuronales, $p(x_i \mid x_{i-(n-1):i-1})$ viene dada por una red neuronal:

$$p(\text{cheese} \mid \text{ate}, \text{the}) = \text{alguna-red-neuronal}(\text{ate}, \text{the}, \text{cheese}).$$

Nótese que la longitud de contexto sigue estando acotada por n , pero ahora es **estadísticamente factible** estimar modelos de lenguaje neuronales para valores mucho mayores de n .

Técnicas principales

- Skip-gram, como el modelo word2vec
- RNN-LSTM: permitió que la distribución condicional de un token x_i dependiera del **contexto completo** $x_{1:i-1}$ (efectivamente $n = \infty$), pero fueron difíciles de entrenar.
- Transformers: arquitectura más reciente (desarrollada para traducción automática en 2017) que volvió a tener longitud de contexto fija n , pero fueron mucho **más fáciles de entrenar** a gran escala.

¿Qué aprende el preentrenamiento?

- Beauchef Campus is located in _____, Chile. [Trivia]
- I put ____ fork down on the table. [syntax]
- The woman walked across the street, checking for traffic over ____ shoulder. [coreference]
- I went to the ocean to see the fish, turtles, seals, and _____. [lexical semantics/topic]
- Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was _____. [sentiment]
- Iroh went into the kitchen to make some tea. Standing next to Iroh, Zuko pondered his destiny. Zuko left the _____. [some reasoning – this is harder]
- I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21, _____. [some basic arithmetic; they don't learn the Fibonacci sequence]

Outline : Habilidades y Aprendizaje en Contexto

Introducción

**Habilidades y Aprendizaje en
Contexto**

Tokenización y Entrenamiento

Instrucciones y Alineamientos

Razonamientos

Entrenamiento en la práctica

Evaluación de LLMs

LLMs como Agentes

Rendimiento de GPT-3 en TriviaQA [19]

Tarea: dada una pregunta de trivia, generar la respuesta.

Adaptación Definimos un prompt basado en las instancias de entrenamiento (si las hay) y la pregunta, y tomamos la completación como la respuesta predicha (demo):

Q: 'Nude Descending A Staircase' es quizás la pintura más famosa de qué artista del siglo XX?

A: Marcel Duchamp

Modelo	Precisión (%)
RAG	68.0
GPT-3 (zero-shot)	64.3
GPT-3 (few-shot)	71.2

Rendimiento de GPT-3 en Traducción

Tarea: traducir una oración en un idioma fuente (p. ej., alemán) a una oración en un idioma objetivo (p. ej., inglés)

Modelo	Precisión (%)
SOTA (supervisado)	40.2
GPT-3 (zero-shot)	27.2
GPT-3 (few-shot)	40.6

- ¡Incluso sin datos de entrenamiento supervisados, GPT-3 iguala el estado del arte de un sistema totalmente supervisado!
- Los resultados de inglés a un idioma extranjero son mucho peores.

Habilidades emergentes: Aprendizaje Zero- y Few-Shot

- **Zero-shot learning:** Resolver tareas sin ningún ejemplo específico.
- **Few-shot learning:** Alcanzar alto rendimiento con ejemplos mínimos [5].
- **Aprendizaje en contexto (In-context learning):** GPT-3 usa el contexto provisto en la entrada, sin actualizar los pesos del modelo.

Habilidades emergentes: Aprendizaje Zero- y Few-Shot

- **Zero-shot learning:** Resolver tareas sin ningún ejemplo específico.
 - **Few-shot learning:** Alcanzar alto rendimiento con ejemplos mínimos [5].
 - **Aprendizaje en contexto (In-context learning):** GPT-3 usa el contexto provisto en la entrada, sin actualizar los pesos del modelo.
- ⇒ Los modelos resuelven tareas basándose en secuencias de entrada y no en actualizar sus pesos de entrenamiento, un cambio respecto a los paradigmas tradicionales de entrenamiento.

Zero- y Few-shot

The three settings we explore for in-context learning

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



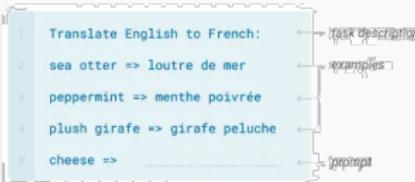
One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Traditional fine-tuning (not used for GPT-3)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



Habilidades emergentes

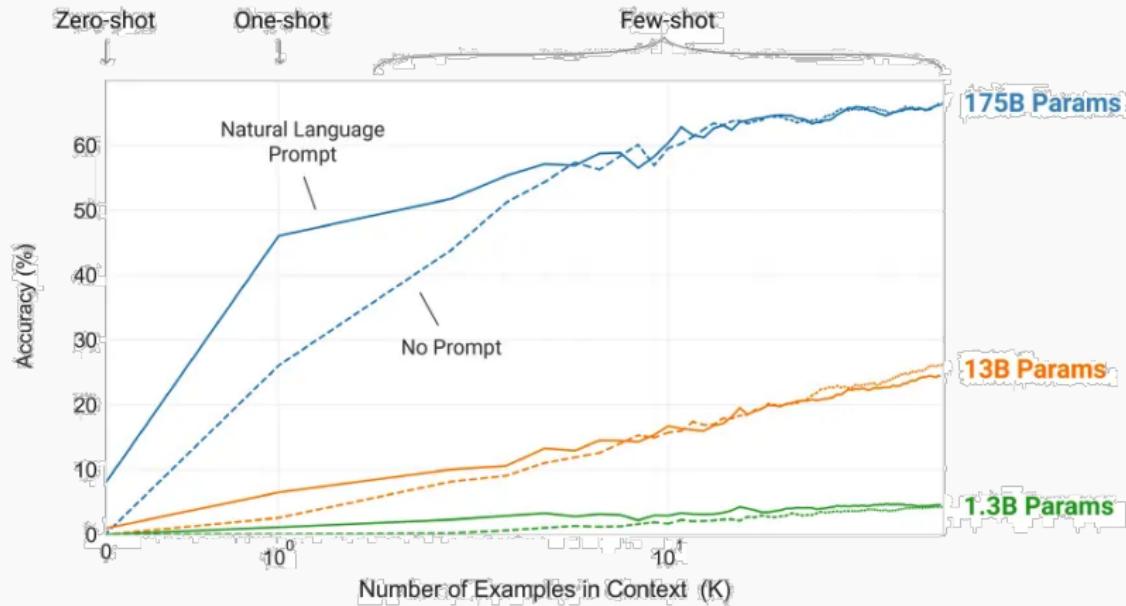
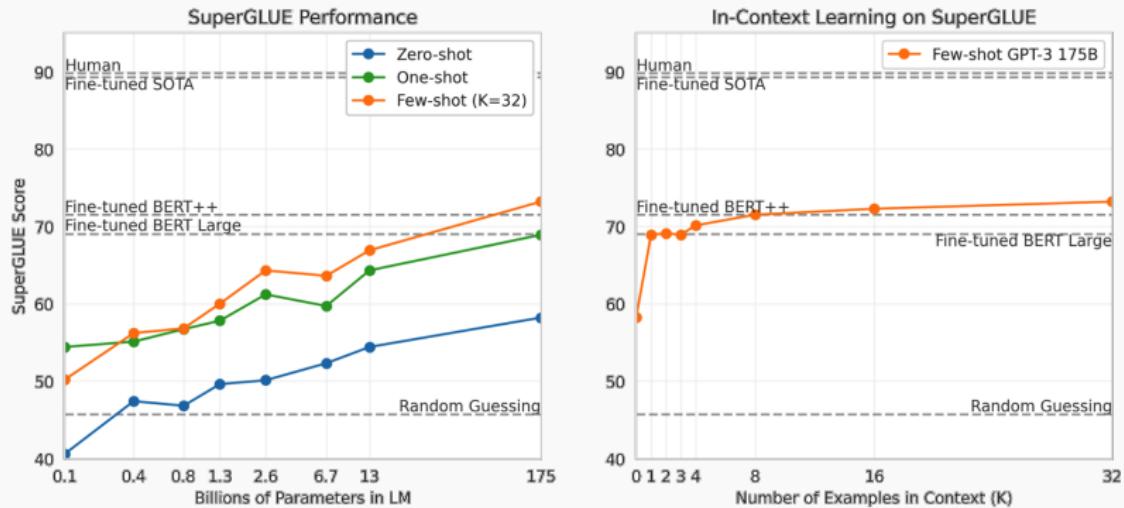


Figure 2: Una tarea simple que requiere que el modelo elimine símbolos extraneos de una palabra.

A partir de cierto tamaño, el modelo es capaz de abordar muchas tareas no vistas explicitamente.

Habilidades Zero- y Few-shot



Las prestaciones son mejores con modelos más grandes y usando más ejemplos.

Habilidades emergentes

Definición de [59]

Una habilidad se considera emergente si no está presente en modelos pequeños pero sí en modelos grandes. Así, **las habilidades emergentes no pueden predecirse simplemente extrapolando el rendimiento de modelos más pequeños.**

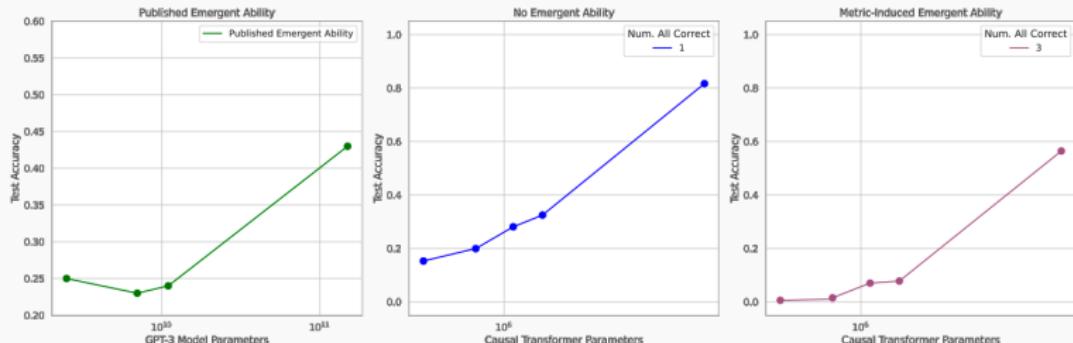
Habilidades emergentes

Definición de [59]

Una habilidad se considera emergente si no está presente en modelos pequeños pero sí en modelos grandes. Así, **las habilidades emergentes no pueden predecirse simplemente extrapolando el rendimiento de modelos más pequeños.**

Pero en realidad.. habilidades emergentes son una ilusión? [43]

Este comportamiento se debe mas a la **elección de la métrica** que a cambios fundamentales en el comportamiento del modelo con la escala! Con métricas adecuadas, las habilidades **aparecen de forma continua!**



Outline : Tokenización y Entrenamiento

Introducción

Habilidades y Aprendizaje en
Contexto

Tokenización y Entrenamiento

Instrucciones y Alineamientos

Razonamientos

Entrenamiento en la práctica

Evaluación de LLMs

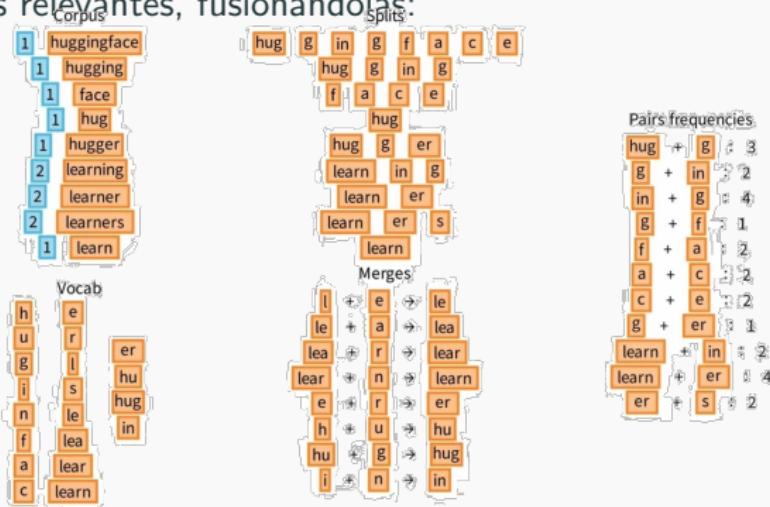
LLMs como Agentes

Tokenización

Tokenización

Convierte el texto en unidades más pequeñas (tokens) para el procesamiento por el modelo (antes de crear embeddings). Es un paso crucial que **afecta el rendimiento y el tiempo de procesamiento**.

Ejemplo: GPT-3 usa Byte Pair Encoding (BPE; [45]), un **tokenizador aprendido a partir de datos** eficiente para encontrar las unidades básicas más relevantes, fusionándolas:



Objetivos de Entrenamiento

Los modelos pueden preentrenarse usando varios objetivos, tales como:

- **Modelado de lenguaje autoregresivo**: predecir el siguiente token dado el contexto previo.
- **Modelado de lenguaje enmascarado**: predecir un token enmascarado dentro de una oración.
- **Predicción de la oración siguiente**: predecir si dos oraciones se siguen una a la otra.

Luego se afinan para varios objetivos:

- **Fine-tuning por instrucciones** mejora el rendimiento en tareas de seguimiento de instrucciones [42]
- **Alineamiento del modelo** con las preferencias de los usuarios

Outline : Instrucciones y Alineamientos

Introducción

Habilidades y Aprendizaje en
Contexto

Tokenización y Entrenamiento

Instrucciones y Alineamientos

Razonamientos

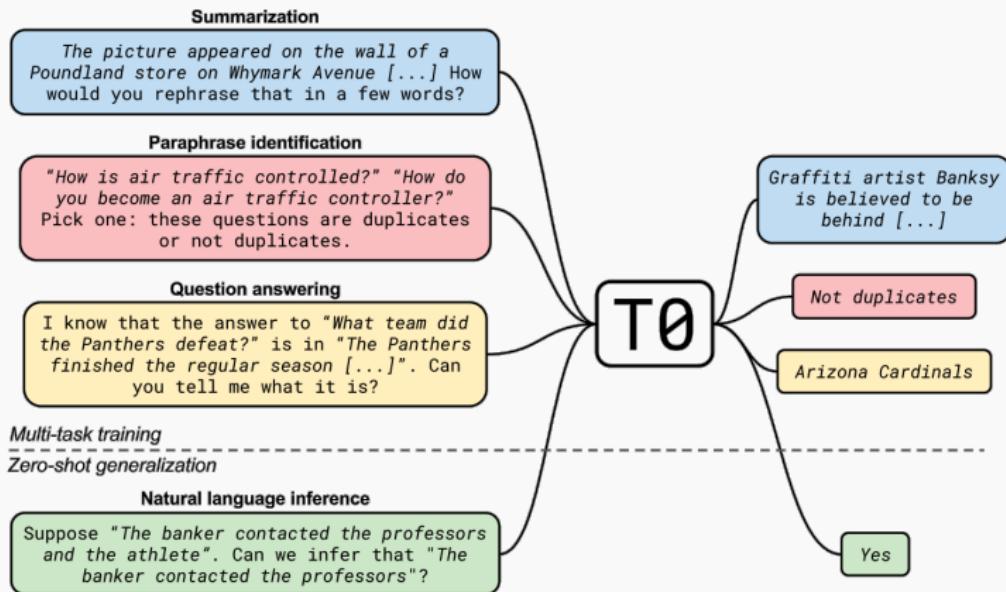
Entrenamiento en la práctica

Evaluación de LLMs

LLMs como Agentes

Instrucciones

- Prompts basados en instrucciones mejoran el rendimiento [58]
- Instrucciones claras y bien estructuradas guían al modelo hacia resultados precisos [9]
- El ajuste por instrucciones ayuda a los modelos a **generalizar mejor en tareas nuevas** [42]



Alineamiento I

- El alineamiento asegura que los modelos se comporten de acuerdo con valores humanos [2]
- **Reinforcement Learning from Human Feedback (RLHF)** es un enfoque para aprender lo que los usuarios desean [33]

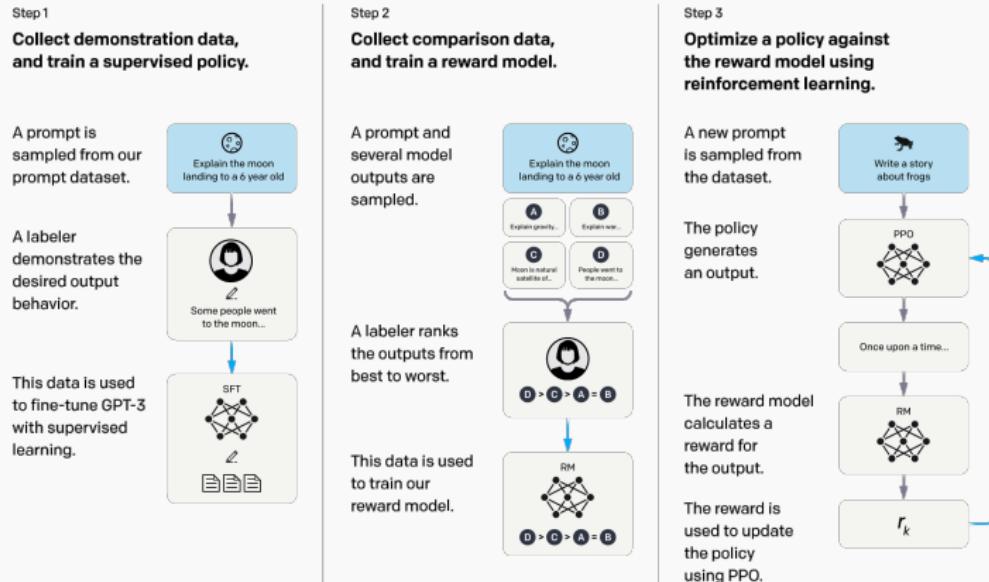


Figure 4: Tres pasos: Aprendizaje Supervisado, Modelo de Recompensa, RLHF

Alineamiento II: RLHF

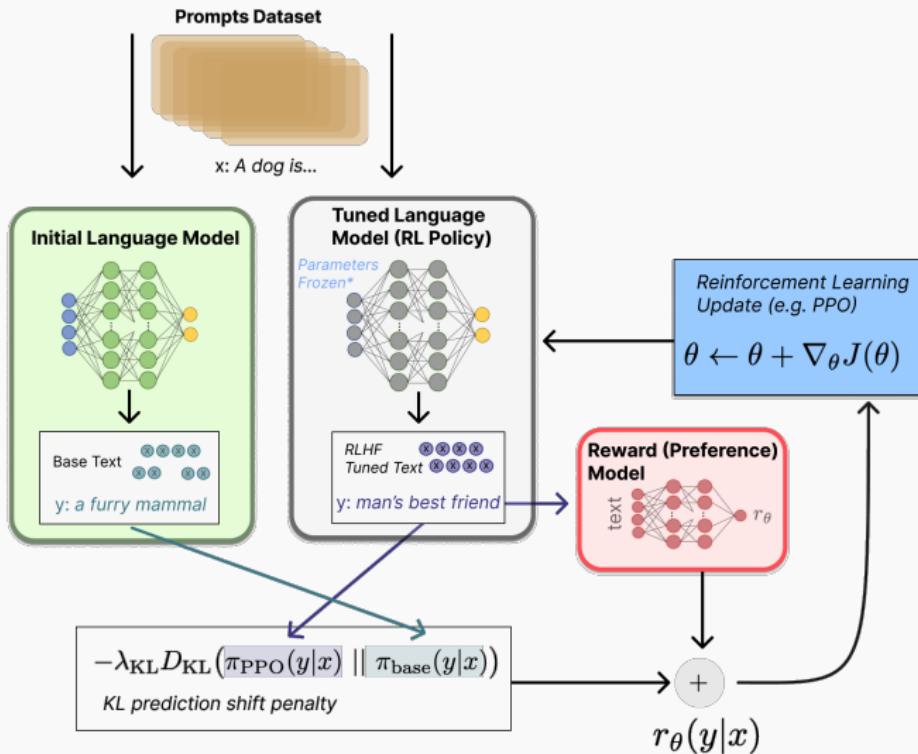


Figure 5: RLHF en detalle. En la realidad, la política RL genera texto, que se introduce en el modelo inicial para producir sus probabilidades relativas para la penalización KL.

Alineamiento III

- En realidad, el alineamiento puede lograrse usando 1,000 ejemplos seleccionados a mano! [68]
- Proximal Policy Optimization (PPO) fue la pérdida original de RLHF, pero existen métodos más eficientes como Direct Preference Optimization (DPO) [40]
- Procesos para razonar usando pasos de pensamiento internos como Thought Preference Optimization (TPO) [61]

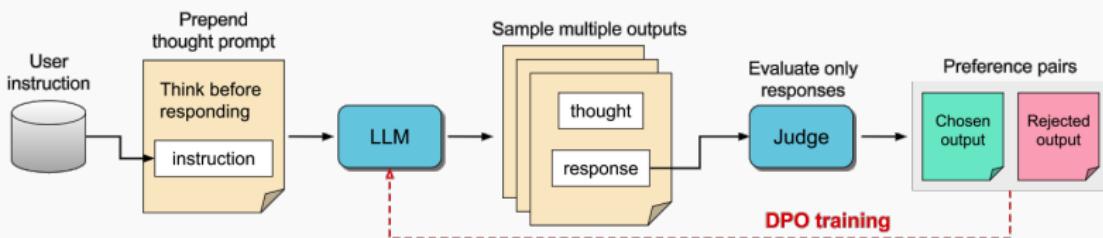


Figure 1: Thought Preference Optimization: We start by prompting the LLM to generate thoughts before its response. After sampling different outputs, we feed the response parts to the judge model which determines the best and worst ones. Then we use the corresponding full outputs as chosen and rejected pairs for DPO optimization. We perform multiple iterations of this training.

Alineamiento IV

¿Y las Preferencias divergentes? [66]

- En algunos temas subjetivos, los anotadores pueden no estar de acuerdo
- Es importante que un modelo general refleje la subjetividad y la diversidad en sus respuestas
- Es difícil para los LLM reflejar las opiniones de las minorías

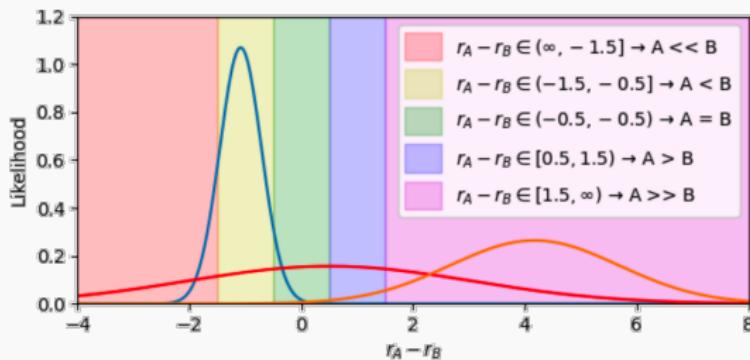


Figure 6: Reward models tienen diferentes modos al observar el pdf (del KL)

Outline : Razonamientos

Introducción

Habilidades y Aprendizaje en
Contexto

Tokenización y Entrenamiento

Instrucciones y Alineamientos

Razonamientos

Entrenamiento en la práctica

Evaluación de LLMs

LLMs como Agentes

Cadena de pensamiento (Chain of Thought) [60]

- Los prompts de Chain of Thought (CoT) ayudan a los modelos a descomponer tareas complejas en pasos, mostrando simplemente respuestas de razonamiento en múltiples pasos con ICL
- Útil para tareas de razonamiento en varios pasos.
- CoT mejora rendimiento en aritmética, lógica y razonamiento.

Standard Prompting	Chain-of-Thought Prompting
<p>Model Input</p> <p>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?</p> <p>A: The answer is 11.</p> <p>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</p>	<p>Model Input</p> <p>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?</p> <p>A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.</p> <p>Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?</p>
<p>Model Output</p> <p>A: The answer is 27. X</p>	<p>Model Output</p> <p>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. ✓</p>

Figure 7: Aumentar con un razonamiento permite un mejor Few-Shot

Chain of Thought Zero-Shot [21]

- El modelo genera CoT sin ejemplos específicos
- Ha mostrado un rendimiento notable en tareas de razonamiento sin entrenamiento específico para la tarea.
- Permite una generalización efectiva en escenarios complejos.

(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. **X**

(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 **X**

(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. **✓**

Chain of Thought Zero-Shot [21]

- El modelo genera CoT sin ejemplos específicos
- Ha mostrado un rendimiento notable en tareas de razonamiento sin entrenamiento específico para la tarea.
- Permite una generalización efectiva en escenarios complejos.

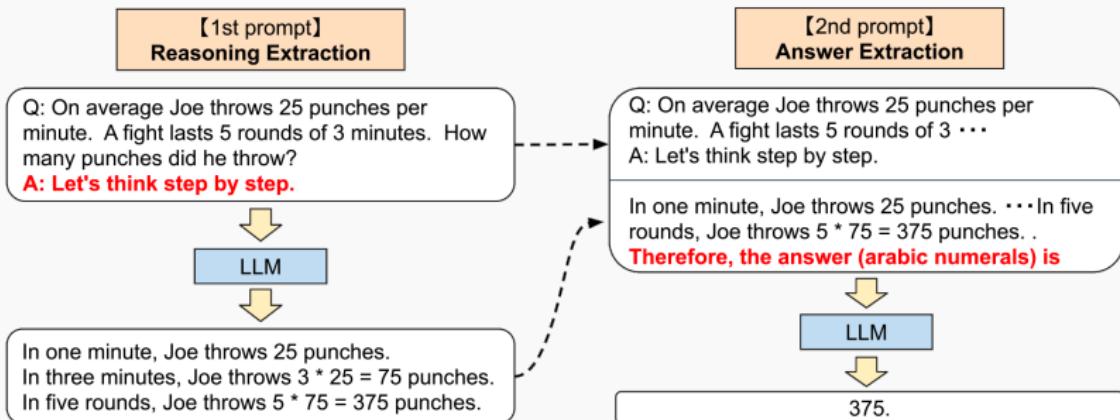


Figure 8: El LLM no genera toda la respuesta de una vez

Existen técnicas para encontrar el mejor prompt [62]: "Take a deep breath a work on this problem step-by-step"

Decodificación CoT [55]

I have 3 apples, my dad has 2 more apples than me,
how many apples do we have in total?



Was Nicolas Cage born in an even or odd year?

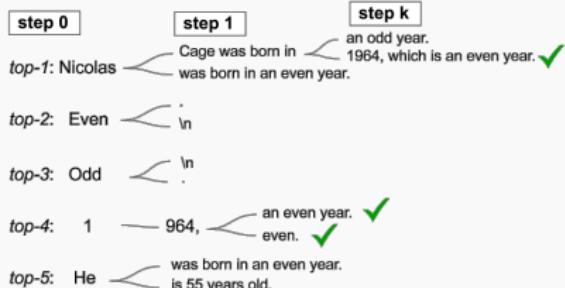


Figure 9: Método de decodificación para obtener texto generado con las mejores explicaciones

- En lugar de la decodificación codiciosa convencional, se investigan las alternativas top- k de tokens, descubriendo que las trayectorias CoT con frecuencia están presentes en estas secuencias.
- La presencia de una CoT en la ruta de decodificación se correlaciona con una mayor confianza en la respuesta decodificada por el modelo.
- **Habilidades intrínsecas de razonamiento de los LLM**

Otros métodos de razonamiento

- Prompting de autorreflexión: el modelo reflexiona sobre su respuesta inicial y la mejora [48]
- Tree-of-Thoughts: Extiende Chain of Thought permitiendo pasos de razonamiento ramificados [64]
- Self-Ask: Anima al modelo a hacerse preguntas aclaratorias para descomponer problemas [38]
- Deliberación democrática: Los modelos debaten las salidas de los demás, conduciendo a consenso o razonamiento corregido [3]
- Marcos basados en agentes: Múltiples agentes corrigen o refinan el razonamiento de los demás, mejorando la calidad final.
- Thoughts Preference Optimization [61]

Outline : Entrenamiento en la práctica

Introducción

Habilidades y Aprendizaje en
Contexto

Tokenización y Entrenamiento

Instrucciones y Alineamientos

Razonamientos

Entrenamiento en la práctica

Evaluación de LLMs

LLMs como Agentes

Conjuntos de entrenamiento: cada vez más tokens

- Datos de entrenamiento son el combustible del LLM
- Textos raspados de la web o subtítulos, pero también contenido de calidad de textbooks, artículos científicos, textos legales, libros, Wiki, código, ...
- Las empresas privadas no suelen abrir sus conjuntos de train ya que impacta fuertemente el rendimiento del modelo

Component	Raw Size	Weight	Epochs	Effective Size	Mean Document Size
Pile-CC	227.12 GiB	18.11%	1.0	227.12 GiB	4.33 KiB
PubMed Central	90.27 GiB	14.40%	2.0	180.55 GiB	30.55 KiB
Books ^{3†}	100.96 GiB	12.07%	1.5	151.44 GiB	538.36 KiB
OpenWebText2	62.77 GiB	10.01%	2.0	125.54 GiB	3.85 KiB
ArXiv	56.21 GiB	8.96%	2.0	112.42 GiB	46.61 KiB
Github	95.16 GiB	7.59%	1.0	95.16 GiB	5.25 KiB
FreeLaw	51.15 GiB	6.12%	1.5	76.73 GiB	15.06 KiB
Stack Exchange	32.20 GiB	5.13%	2.0	64.39 GiB	2.16 KiB
USPTO Backgrounds	22.90 GiB	3.65%	2.0	45.81 GiB	4.08 KiB
PubMed Abstracts	19.26 GiB	3.07%	2.0	38.53 GiB	1.30 KiB
Gutenberg (PG-19) [†]	10.88 GiB	2.17%	2.5	27.19 GiB	398.73 KiB
OpenSubtitles [†]	12.98 GiB	1.55%	1.5	19.47 GiB	30.48 KiB
Wikipedia (en) [†]	6.38 GiB	1.53%	3.0	19.13 GiB	1.11 KiB
DM Mathematics [†]	7.75 GiB	1.24%	2.0	15.49 GiB	8.00 KiB
Ubuntu IRC	5.52 GiB	0.88%	2.0	11.03 GiB	545.48 KiB
BookCorpus2	6.30 GiB	0.75%	1.5	9.45 GiB	369.87 KiB
EuroParl [†]	4.59 GiB	0.73%	2.0	9.17 GiB	68.87 KiB
HackerNews	3.90 GiB	0.62%	2.0	7.80 GiB	4.92 KiB
YoutubeSubtitles	3.73 GiB	0.60%	2.0	7.47 GiB	22.55 KiB
PhilPapers	2.38 GiB	0.38%	2.0	4.76 GiB	73.37 KiB
NIH ExPorter	1.89 GiB	0.30%	2.0	3.79 GiB	2.11 KiB
Enron Emails [†]	0.88 GiB	0.14%	2.0	1.76 GiB	1.78 KiB
The Pile	825.18 GiB			1254.20 GiB	5.91 KiB

Figure 10: Composición de The Pile [13]

Training Datasets: Ejemplos de conjuntos abiertos recientes

- **C4**: Colossal Clean Crawled Corpus [11]: 0.3 TB
- **The Pile** [13]: 0.8 TB
- **Dolma**: un corpus abierto de tres billones de tokens para investigación de preentrenamiento [50]: 4.5 TB
- **The FineWeb Datasets** [35]: 93.4 TB
- **RedPajama**: Conjunto abierto para entrenar LLMs [57]: 270 TB

Table 1: Comparison of open pretraining Datasets along the dimensions of transparency, versatility, and scale.

Dataset	Transparency		Versatility			Scale (TB)
	Open Access	Open Code	Raw Data	Composite	Multilingual	
Refined Web [44]	✓(subset)	✗	✗	✗	✗	2.8
FineWeb [43]	✓	✓	✗	✗	✗	93.4
FineWeb-EDU [43]	✓	✓	✗	✗	✗	8.8
C4 [46]	✓	✓	✗	✗	✗	0.3
mC4 [63]	✓	✓	✗	✗	✓	9.7
DCLM baseline [30]	✓	✓	✗	✗	✗	10.0
DCLM-Pool [30]	✓	✓	✓	✗	✓	340.0
Dolma v1.7 [52]	✓	✓	✗	✓	✗	4.5
Pile [17]	✓	✓	✗	✓	✗	0.8
SlimPajama [51]	✓	✓	✗	✓	✗	0.9
ROOTS [26, 27]	✓	✓	✗	✓	✓	1.6
RedPajama-V1	✓	✓	✗	✓	✗	3.0
RedPajama-V2	✓	✓	✓	✗	✓	270.0

Training Datasets: Ejemplos de conjuntos abiertos recientes

- **C4**: Colossal Clean Crawled Corpus [11]: 0.3 TB
- **The Pile** [13]: 0.8 TB
- **Dolma**: un corpus abierto de tres billones de tokens para investigación de preentrenamiento [50]: 4.5 TB
- **The FineWeb Datasets** [35]: 93.4 TB
- **RedPajama**: Conjunto abierto para entrenar LLMs [57]: 270 TB

Table 1: Comparison of open pretraining Datasets along the dimensions of transparency, versatility, and scale.

Dataset	Transparency		Versatility			Scale (TB)
	Open Access	Open Code	Raw Data	Composite	Multilingual	
Refined Web [44]	✓(subset)	✗	✗	✗	✗	2.8
FineWeb [43]	✓	✓	✗	✗	✗	93.4
FineWeb-EDU [43]	✓	✓	✗	✗	✗	8.8
C4 [46]	✓	✓	✗	✗	✗	0.3
mC4 [63]	✓	✓	✗	✗	✓	9.7
DCLM baseline [30]	✓	✓	✗	✗	✗	10.0
DCLM-Pool [30]	✓	✓	✓	✗	✓	340.0
Dolma v1.7 [52]	✓	✓	✗	✓	✗	4.5
Pile [17]	✓	✓	✗	✓	✗	0.8
SlimPajama [51]	✓	✓	✗	✓	✗	0.9
ROOTS [26, 27]	✓	✓	✗	✓	✓	1.6
RedPajama-V1	✓	✓	✗	✓	✗	3.0
RedPajama-V2	✓	✓	✓	✗	✓	270.0

Conjuntos de entrenamiento: Limpieza antes de usar I

Data raspada de la web es bastante ruidosa y hay que limpiarla:

Ruido de caracteres: "And it was"

Marcas, rupturas de sintaxis, etc... todo eso que produce texto no natural es perjudicial y puede impedir la convergencia.

Diversificación: "~~Yes lol bro!~~ — ~~Yes bro!~~"

Diversificación mejora el pre-training eliminando ejemplos inútiles (simples) [53].¹ PPL puede ayudar a filtrar doc simples.

Desduplicación: mismo contenido en diferentes páginas

Se ha identificado como factor importante para mejorar los modelos de lenguaje [23]. Los datos repetidos han demostrado ser **cada vez más perjudiciales para la calidad del modelo a medida que aumenta el número de parámetros** [16]:

- para un modelo de 1B parámetros, cien duplicados son perjudiciales;
- a 175B, unos duplicados pueden tener efecto desproporcionado.

¹Algunos también usan datos diversos de buena calidad de libros de texto [14]

Conjuntos de entrenamiento: Limpieza antes de usar II

¿Concretamente?

- API para extraer texto eficientemente desde XML (trafilatura; [4]), eliminar tablas,...
- Revisar marcas que puedan romper la extracción
- Eliminar líneas poco informativas con menos de unos pocos tokens
- Desduplicación: Fuzzy usando minHash, exacta usando n-gramas con la librería text-dedup [20].
- Usar **heurísticos simples ayuda mucho** [46]
- Ejemplos concretos: lo que hicieron para [RedPajama](#)

DOCUMENT PREPARATION			FILTERING		DEDUPLICATION	
URL filtering	Text extraction	Language identification	Document-wise filtering	Line-wise filtering	Deduplication	URL deduplication
Aggregated block-list, URL scoring, common HQ sources blocked Appendix G.1	From WARC using warcio, trafilatura for extraction Barbaresi (2021)	fastText classifier from CCNet, thresholding on top language score Wenzek et al. (2020)	In-document repetition removal and quality heuristics from MassiveWeb Rae et al. (2021)	Remove undesirable lines (call to actions, navigation buttons, social counters, etc.) Appendix G.2	Fuzzy deduplication w/ MinHash + exact substring deduplication w/ suffix arrays Lee et al. (2022)	Remove URLs revisited across Common-Crawl dumps Section 3.3

Figure 11: Ejemplo de limpieza de corpus [36, 35]

Conjuntos de ajuste por instrucciones: FLAN [58, 29, 9]

Finetuning tasks

TO-SF

Commonsense reasoning
Question generation
Closed-book QA
Adversarial QA
Extractive QA
Title/context generation
Topic classification
Struct-to-text
...

55 Datasets, 14 Categories,
193 Tasks

Muffin

Natural language inference
Code instruction gen.
Program synthesis
Dialog context generation
Closed-book QA
Conversational QA
Code repair
...

69 Datasets, 27 Categories, 80 Tasks

CoT (Reasoning)

Arithmetic reasoning
Commonsense Reasoning
Implicit reasoning
Explanation generation
Sentence composition
...

9 Datasets, 1 Category, 9 Tasks

Natural Instructions v2

Cause effect classification
Commonsense reasoning
Named entity recognition
Toxic language detection
Question answering
Question generation
Program execution
Text categorization
...

372 Datasets, 108 Categories,
1554 Tasks

- ❖ A Dataset is an original data source (e.g. SQuAD).
- ❖ A Task Category is unique task setup (e.g. the SQuAD dataset is configurable for multiple task categories such as extractive question answering, query generation, and context generation).
- ❖ A Task is a unique <dataset, task category> pair, with any number of templates which preserve the task category (e.g. query generation on the SQuAD dataset.)

Held-out tasks

MMLU

Abstract algebra
College medicine
Professional law
Sociology
Philosophy
...

57 tasks

BBH

Boolean expressions
Tracking shuffled objects
Dyck languages
Navigate
Word sorting
...

27 tasks

TyDiQA

Information seeking QA
8 languages

MGSM

Grade school math problems
10 languages

Figure 12: Modelos de lenguaje afinados con anotaciones (FLAN)

Conjuntos recientes de ajuste por instrucciones

Algunos ejemplos de conjuntos de instrucciones:

- **Alpaca**: 52k pares instrucción-respuesta completamente **generados por máquina**. Tareas simples como generación de texto, resumen y preguntas y respuestas. [52]
- **DollyV2**: Conjunto abierto **curado manualmente**, enfatizando tareas diversas con 15k pares [10].
- **Open Instruction Generalist (OIG)**: **43M instrucciones** de 30 conjuntos, 75% de fuentes académicas (p. ej., P3, FLAN) y 25% tareas diversas como código, poesía y resumen [22]
- **Aya Dataset**: Primer conjunto abierto de instrucciones en **idiomas diversos y no solo inglés** (3.5% del conjunto) [49]

Outline : Evaluación de LLMs

Introducción

Habilidades y Aprendizaje en
Contexto

Tokenización y Entrenamiento

Instrucciones y Alineamientos

Razonamientos

Entrenamiento en la práctica

Evaluación de LLMs

LLMs como Agentes

Cómo evaluarlos

Bueno para muchas cosas: Benchmarks multitarea

- Comenzaron con unas pocas decenas de tareas con BERT
- Ahora más de 200 tareas diversas
- Temas variados, incluso preguntas a nivel universitario, ..

Bueno en conversaciones: Benchmarks multironda

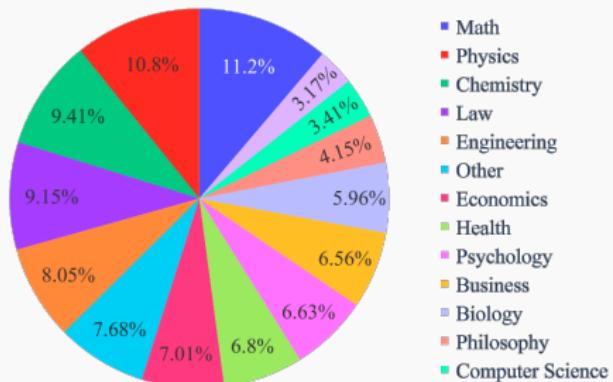
- Evalúan los modelos en un entorno conversacional
- Útil para chatbots o asistentes

¿Pero qué es bueno y qué no?: Humano- y LLM-como-juez

- Usando el diálogo completo para juzgar.
- Forma automática: usar LLMs más grandes para evaluar a otros.
- El LLM correlaciona bastante bien con humanos, sin embargo introduce algunos sesgos.

Benchmarks multitarea

- Los benchmarks multitarea evalúan LLMs en una amplia gama de tareas de distintos dominios, algunos de ellos son:
 - **MMLU** (Massive Multitask Language Understanding) [15] y **MMLU-Pro** [56]: 57+ tareas cubriendo historia, ciencia, derecho,...
 - **BIG-Bench** [51]: Más de 200 tareas diseñadas para medir capacidades diversas incluyendo razonamiento, ética y humor.
 - **TruthfulQA** [28]: Se centra en evaluar la veracidad de las salidas del modelo en varios dominios (si dice disparates o no)
 - **SciBench** [54]: Problemas científicos a nivel universitario
- El objetivo es probar la generalización del modelo en dominios no relacionados.
- Destaca las fortalezas y debilidades de los modelos en áreas específicas (p. ej., razonamiento vs. conocimiento factual).



(a) Distribution of Disciplines in MMLU-Pro 44

Multi-turn Benchmarks: LLM como agentes conversacionales

- Evalúan la habilidad del modelo para manejar conversación y retener contexto a lo largo de múltiples intercambios:
 - **MT-Bench** [67]: Un benchmark de dos rondas que evalúa calidad conversacional, coherencia y retención de contexto.
 - **MT-Bench-101** [1]: Benchmark multironda, diseñado para probar diálogos complejos y extendidos.
 - Benchmarks clásicos de ConvQA [17, 8, 41, 32, 18]: Prueban la capacidad del modelo de responder manteniendo el contexto.
- Una primera forma simple de integrar interacciones en la evaluación del chatbot
- Mantener el contexto y ofrecer respuestas coherentes en conversación es crucial para asistentes conversacionales



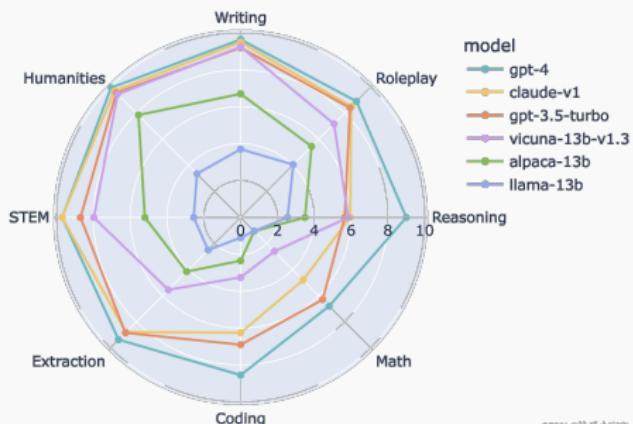
Figure 13: Taxonomía MT-Bench-101

Evaluación de interacción: Humano- o LLM-como-juez

- **¿Por qué diferente?** En un diálogo, a veces hay muchas respuestas correctas posibles, y los benchmarks anteriores fallan en evaluarlo.
- **Solución:** Un humano (costoso) o LLMs se usan a veces como evaluadores de sus propias salidas o de otros modelos.

Juez humano

Chatbot Arena [67, 7]: Plataforma de crowdsourcing donde los usuarios comparan LLMs votando su desempeño en interacciones de chat en vivo, cara a cara.



Evaluación de interacción: Humano- o LLM-como-juez

- **¿Por qué diferente?** En un diálogo, a veces hay muchas respuestas correctas posibles, y los benchmarks anteriores fallan en evaluarlo.
- **Solución:** Un humano (costoso) o LLMs se usan a veces como evaluadores de sus propias salidas o de otros modelos.

Juez LLM [67]

- SoTA LLM pueden igualar preferencias humanas en entornos controlados y crowd-sourced ($\approx 80\%$ de acuerdo, igual que los humanos).
- Útil cuando la evaluación humana es costosa o imposible, aunque existen desafíos con sesgos y objetividad en los juicios de LLM.
- Puede usarse incluso sin el contexto de la conversación.

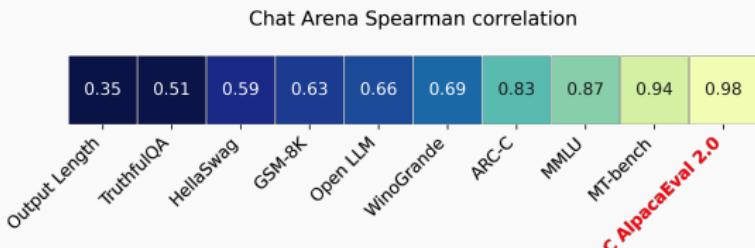


Figure 14: AlpacaEval [12, 26] correlaciona alto con humanos en ChatGPT

Otros estudios sobre LLM, menos centrados en el rendimiento

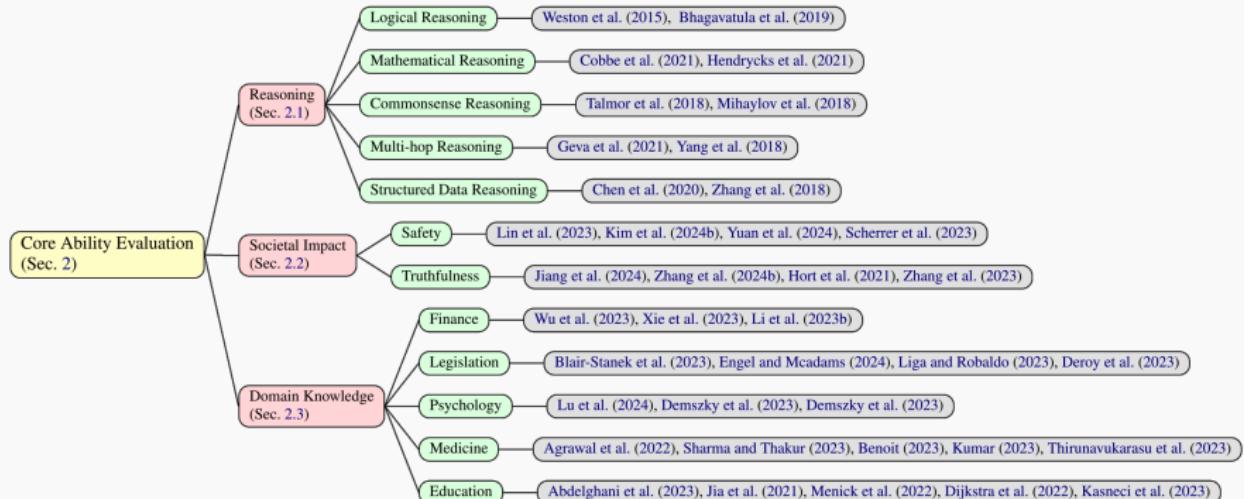


Figure 15: Visión general de evaluaciones de habilidades centrales de [37]

Los LLM también pueden afinarse para juzgar

Metric	MT	TS	DG	IC	D2T	SG	GE	REF	LLMs	Protocol	Aspects
<i>Prompt-based Evaluation</i>											
BARTScore (Yuan et al., 2021)	✓	✓	*	○	✓	*	*	✓	BART	Prob	CON/COH/REL/FLU/ INF/COV/ADE
GPTScore (Fu et al., 2023)	✓	✓	✓		✓	*	*		GPT3	Prob	CON/COH/REL/FLU/COV/ MQM/INF/FAC/INT/ENG/NAT
G-EVAL (Liu et al., 2023c)	*	✓	✓		*	*	*		ChatGPT/GPT-4	Advanced	CON/COH/REL/FLU/ /NAT/ENG/GRO
ICE (Jain et al., 2023)	*	✓	*		*	*	*		GPT-3	Score	CON/COH/REL/FLU
GEMBA (Kocmi and Federmann, 2023)	✓	*	*		*	*	*		ChatGPT	Score/Likert	NONE
LLM_eval (Chiang and Lee, 2023)	*	*	*		*	✓	*		ChatGPT	Likert	GRAM/COH/REL/LIK
FairEval (Wang et al., 2023c)	*	*	*		*	*	*		ChatGPT/GPT-4	Pairwise	NONE
AuPEL (Wang et al., 2023e)	*	*	*		*	*	*		Pal-M-2	Pairwise	PER/QUA/REL
DRPE (Wu et al., 2023a)	*	✓	*	*	*	*	*	*	GPT-3	Ensemble	CON/COH/REL/FLU/INT/USE
ChatEval (Chan et al., 2023)	*	*	✓		*	*	*		ChatGPT/GPT-4	Ensemble	NAT/COH/ENG/GRO
WideDeep (Zhang et al., 2023)	*	*	*		*	*	*		ChatGPT	Ensemble	COH/REL/HARM/ACC
PRD (Li et al., 2023c)	*	*	*		*	*	✓		GPT-4/GPT-3.5 Vicuna/Claude/Bard	Ensemble	INF/COH
FACTSCORE (Min et al., 2023)	*					✓			ChatGPT	Advanced	FAC
EAprompt (Li et al., 2023)	✓	*	*		*	*	*		ChatGPT/text-davinci-003	Advanced	NONE
AUTOCALIBRATE (Liu et al., 2023f)	*	✓	*		*	*	*		GPT-4	Likert	CON/COH/REL/FLU/INF/NAT
ALLURE (Hasanbeig et al., 2023)	*	✓	*		*	*	✓		GPT-4	Advanced	CON/COH/FLU/REL
<i>Tuning-based Evaluation</i>											
PRISM (Thompson and Post, 2020)	✓	○	*	○	*	*	*	✓	Transformer	Prob	NONE
TSScore (Qin et al., 2022)	✓	✓	*	*	*	*	*	✓	T5	Prob	NONE
TrueTeacher (Gekhman et al., 2023)	*	✓	*		*	*	*		T5	Likert	CON
X-EVAL (Liu et al., 2023a)	*	✓	✓		✓	*	*		FLAN-T5-large	Likert	DEP/LIK/UND/FLE/INF/INQ INT/SPE/COR/SEM/COH/ENG NAT/GRO/CON/REL/FLU
AUTO-J (Li et al., 2023a)	*	*	*		*	*	*		LLaMA	Likert/Pairwise	ACC/CLL/FEA/CRE/THO STR/LAY/COM/INF
PERSE (Wang et al., 2023a)	*	*	*	*	*	✓	*	✓	LLaMA	Likert/Pairwise	INT/ADA/SUR/CHA/END
PandaLM (Wang et al., 2023f)	*	*	*		*	*	✓		LLaMA	Pairwise	CLA/COM/FOR/ADH
Attscore (Yue et al., 2023)	*	*	*		*	*	✓		Roberta/T5/GPT2 LLaMA/Vicuna	Advanced	CON
TIGERScore (Jiang et al., 2023)	✓	✓	*		✓	✓	✓		LLaMA	Advanced	COH/INF/ACC/COM
INSTRUCTSCORE (Xu et al., 2023)	✓	*	*	*	*	*	*	*	LLaMA	Advanced	NONE
Prometheus (Kim et al., 2023a)	*	*	*		*	*	*		LLaMA-2	Likert/Pairwise	NONE
Prometheus-2 (Kim et al., 2023a)	*	*	*		*	*	*		Mistral 7B	Likert/Pairwise	NONE
CritiqueLLM (Ke et al., 2023)	*	*	*		*	*	*		ChatGLM	Likert	NONE

Figure 16: Métricas de generación de lenguaje natural a partir de LLMs [27]

Una gran familia de LLMs

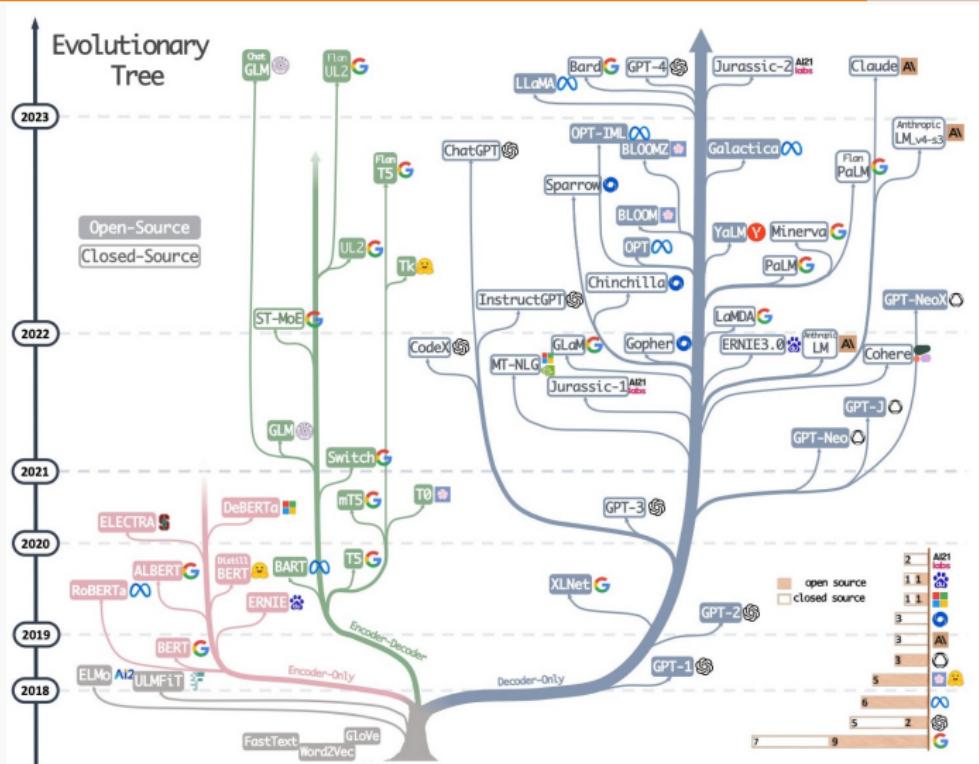


Figure 17: Algunos de los modelos más conocidos; los que están en la misma rama tienen relaciones más cercanas. Más información [aquí](#)

Outline : LLMs como Agentes

Introducción

Habilidades y Aprendizaje en
Contexto

Tokenización y Entrenamiento

Instrucciones y Alineamientos

Razonamientos

Entrenamiento en la práctica

Evaluación de LLMs

LLMs como Agentes

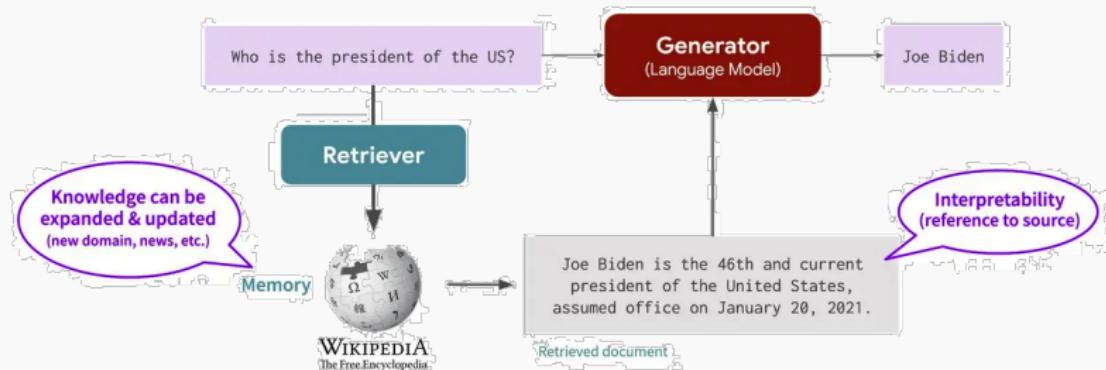
Generación Aumentada por Recuperación (RAG; [24])

RAG integra la **recuperación** desde bases de conocimiento externas con **generación de texto**

Principio

- Los documentos recuperados proporcionan contexto para el modelo, reduciendo la dependencia de la memoria paramétrica estática.
- No confia solo en los parámetros preentrenados pero recupera dinámicamente información relevante desde BD externas.

Retrieval augmentation



Generación Aumentada por Recuperación (RAG; [24])

RAG integra la **recuperación** desde bases de conocimiento externas con **generación de texto**: para agentes buscando conocimiento específico.

Principio

- Los documentos recuperados proporcionan contexto para el modelo, reduciendo la dependencia de la memoria paramétrica estática.
 - No confia solo en los parámetros preentrenados pero recupera dinámicamente información relevante desde BD externas.
-
- **Recuperador (Retriever)**: Identifica los top-k documentos relevantes de un corpus (p. ej., Dense Passage Retrieval, BM25).
 - Puede transformar consultas y documentos en vectores,
 - Emparejarlos después mediante mapeos o medidas de similitud (rápido dado que puede haber MUCHOS documentos).
 - **Generador (Generator)**: Procesa el texto recuperado y la consulta juntos en un único prompt para producir una salida coherente (p. ej., BART, T5).

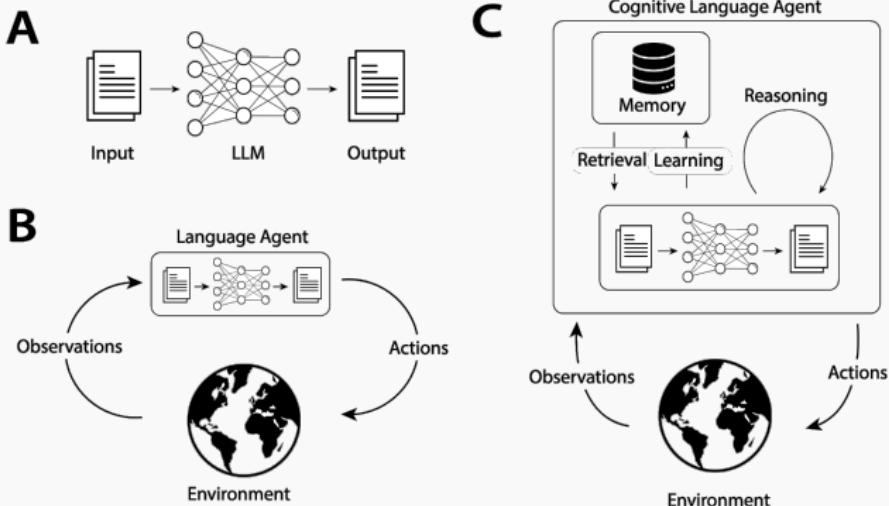
Agentes de Lenguaje

Un agente es una entidad que percibe su entorno a través de sensores y actúa sobre él usando actuadores para lograr objetivos específicos.

Están diseñados para **tomar decisiones, ejecutar acciones de forma autónoma y adaptar su comportamiento** basándose en observaciones o experiencias aprendidas, usando:

- **Memoria:** Almacena contexto e historial de interacción para tareas a largo plazo (puede ser una base de datos de documentos).
- **Simbología:** Representa conocimiento y conceptos como símbolos o reglas estructuradas e interpretables.
- **Razonamiento:** Para procesar entradas y generar salidas accionables.
- **Planificación:** Consiste en un conjunto de reglas, cada una especificando una precondición y una acción, para alcanzar un objetivo global.
- **Interacción con el entorno:** APIs o herramientas que permiten interactuar con sistemas externos.

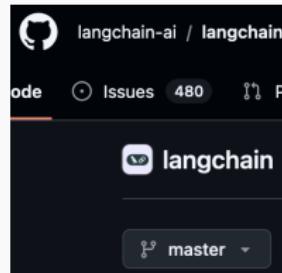
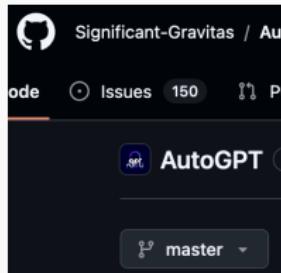
Agentes de Lenguaje: Diferentes usos de los LLMs [63]



- A PNL (NLP):** El LLM toma texto como entrada y produce texto como salida.
- B Agentes de lenguaje:** El LLM en un bucle de retroalimentación directo con el entorno externo transformando observaciones en texto y usando el LLM para elegir acciones.
- C Agentes cognitivos de lenguaje:** El LLM gestiona el estado interno del agente mediante procesos como aprendizaje y razonamiento.

Ejemplos de frameworks de Agentes Cognitivos de Lenguaje LLM

- **AutoGPT**: Descompone tareas y ejecuta subtareas de forma autónoma con mínima intervención humana.
- **BabyAGI**: Combina LLMs con memoria y un bucle de gestión de tareas.
- **LangChain**: Permite encadenar prompts de LLM con herramientas externas, APIs y fuentes de datos.



LLMs usando Herramientas: Toolformer [44]

Los LLMs tienen dificultades con funcionalidades básicas, como aritmética o búsquedas factuales. El **uso de herramientas** les permite interactuar con APIs, bases de datos, código,... para mejorar el razonamiento y la ejecución de tareas que requieren acciones.

Principio

- Proporciona acceso a funcionalidades especializadas, como obtener información en vivo, realizar cálculos o consultar servicios.
- Los LLMs interactúan dinámicamente con herramientas, **permitiendo al agente realizar acciones** que impactan el mundo exterior.



Los LLMs tienen dificultades con funcionalidades básicas, como aritmética o búsquedas factuales. El **uso de herramientas** les permite interactuar con APIs, bases de datos, código,... para mejorar el razonamiento y la ejecución de tareas que requieren acciones.

Principio

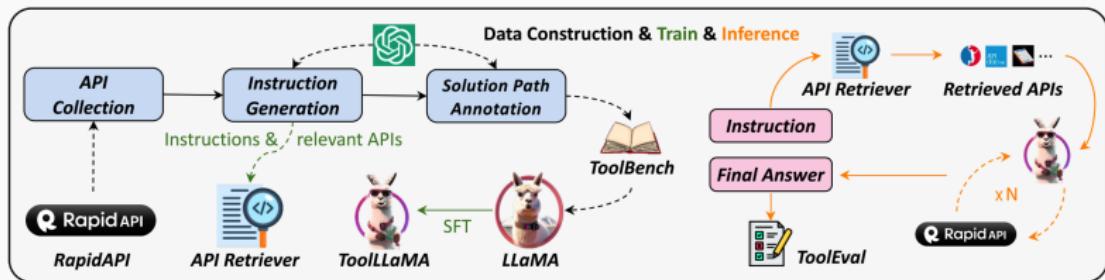
- Proporciona acceso a funcionalidades especializadas, como obtener información en vivo, realizar cálculos o consultar servicios.
 - Los LLMs interactúan dinámicamente con herramientas, **permitiendo al agente realizar acciones** que impactan el mundo exterior.
-
- **Mecanismos clave:**
 - **Los LLMs aprenden a usar las herramientas mediante una plantilla especial** durante el entrenamiento, consultando sistemas externos (p. ej., motores de búsqueda, calculadoras) en LN.
 - **Generación de código:** Los modelos escriben y ejecutan scripts para tareas tal como consultas a BD, simulaciones o cálculos numéricos.
 - **Evaluación:** El benchmark GAIA [31] evalúa razonamiento, manejo

LLMs usando Herramientas

- Los LLMs pueden aprender a usar muchas herramientas en modo zero-shot mediante **conjuntos de datos de fine-tuning orientados al uso de herramientas** [70, 30, 39].
- Las herramientas pueden ser específicas de un dominio, por ejemplo herramientas para el ámbito médico [25].
- Los LLMs también pueden **crear sus propias herramientas**: un LLM como *fabricante de herramientas* y otro como *usuario de herramientas* [6].



TOOLLM: FACILITATING LARGE LANGUAGE MODELS TO MASTER 16000+ REAL-WORLD APIs



LLMs usando herramientas: HuggingGPT [47]

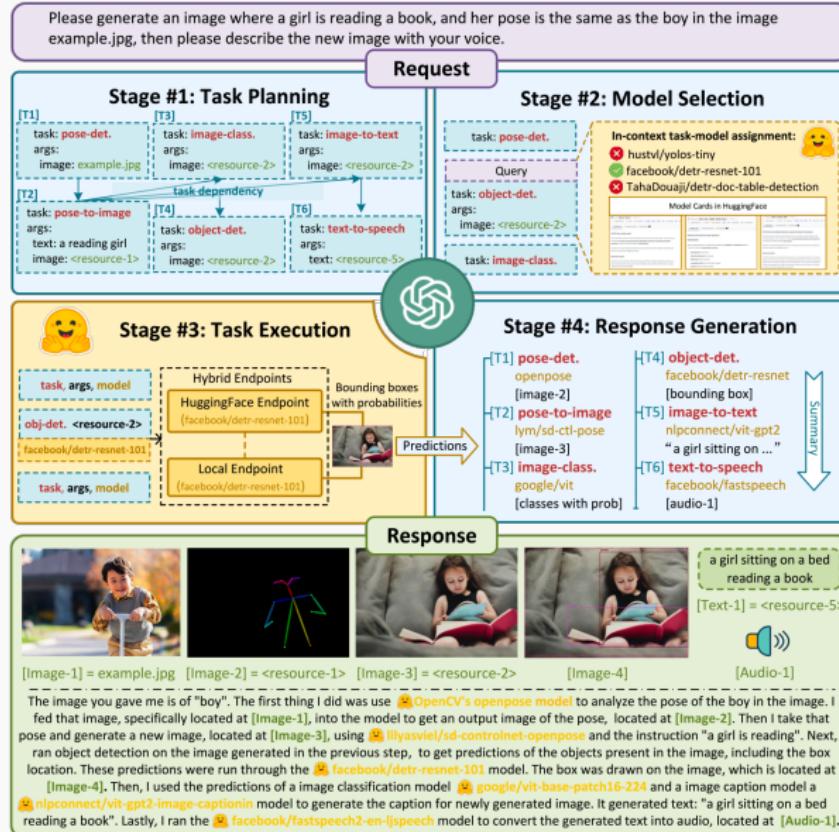
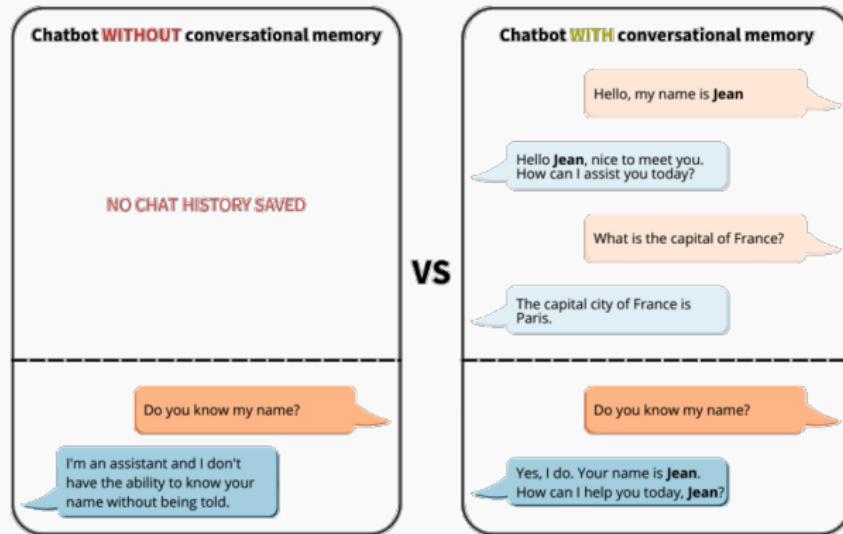


Figure 19: Herramientas tan complejas como modelos en el hub de

Ventana de contexto para chatear: ¿Cómo modelar el historial de conversación?

¿Cómo modelar el historial de conversación?

El modelo transforma la conversación pasada en texto y la coloca en el prompt usando un formato especial, específico para el LLM.



Ventana de contexto para chatear: ¿Cómo modelar el historial de conversación?

¿Cómo modelar el historial de conversación?

El modelo transforma la conversación pasada en texto y la coloca en el prompt usando un formato especial, específico para el LLM.

```
messages = [  
    {"role": "user", "content": "Hi there!"},  
    {"role": "assistant", "content": "Nice to meet you!"},  
    {"role": "user", "content": "Can I ask a question?"}  
]
```

Se pueden usar plantillas de chat con la clase `tokenizer` en `transformers`.

```
tokenizer.apply_chat_template(messages, tokenize=False, add_generation_prompt=True)  
"""<|im_start|>user  
Hi there!<|im_end|>  
<|im_start|>assistant  
Nice to meet you!<|im_end|>  
<|im_start|>user  
Can I ask a question?<|im_end|>  
<|im_start|>assistant  
"""
```

Más información en el tutorial *Chatting with Transformers* aquí.

Plantillas con RAG

```
# Define conversation input
conversation = [
    {"role": "user", "content": "What has Man always dreamed of?"}
]

# Define documents for retrieval-based generation
documents = [
    {
        "title": "The Moon: Our Age-Old Foe",
        "text": "Man has always dreamed of destroying the moon. In this essay, I shall..."
    },
    {
        "title": "The Sun: Our Age-Old Friend",
        "text": "Although often underappreciated, the sun provides several notable benefits..."
    }
]
```

Puede llamarse usando la opción `documents` (si el modelo la soporta) y generar:

```
# Tokenize conversation and documents using a RAG template, returning
# PyTorch tensors.
>>> input_ids = tokenizer.apply_chat_template(
    conversation=conversation,
    documents=documents,
    chat_template="rag",
    tokenize=True,
    add_generation_prompt=True,
    return_tensors="pt").to(device)
# Generate a response
>>> gen_tokens = model.generate(
    input_ids,
    max_new_tokens=100,
    do_sample=True,
    temperature=0.3,
)
# Decode and print the generated text along with generation prompt
>>> gen_text = tokenizer.decode(gen_tokens[0])
```

LLMs usando herramientas

```
def get_current_temperature(location: str, unit: str) -> float:  
    """  
        Get the current temperature at a location.  
  
    Args:  
        location: The location to get the temperature for, in the format "City, Country"  
        unit: The unit to return the temperature in. (choices: ["celsius", "fahrenheit"])  
    Returns:  
        The current temperature at the specified location in the specified units, as a float.  
    """  
    return 22. # A real function should probably actually get the temperature!  
  
list_tools = [get_current_temperature]  
  
messages = [  
    {"role": "system", "content": "You are a bot that responds to weather queries. You should reply  
    ↪ with the unit used in the queried location."},  
    {"role": "user", "content": "Hey, what's the temperature in Paris right now?"}  
]
```

Puede llamarse usando la opción tools:

```
>>> inputs = tokenizer.apply_chat_template(messages, tools=list_tools,  
    ↪ add_generation_prompt=True, return_dict=True, return_tensors="pt")  
>>> inputs = {k: v.to(model.device) for k, v in inputs.items()}  
>>> out = model.generate(**inputs, max_new_tokens=128)  
>>> print(tokenizer.decode(out[0][len(inputs["input_ids"])[0]):]))
```

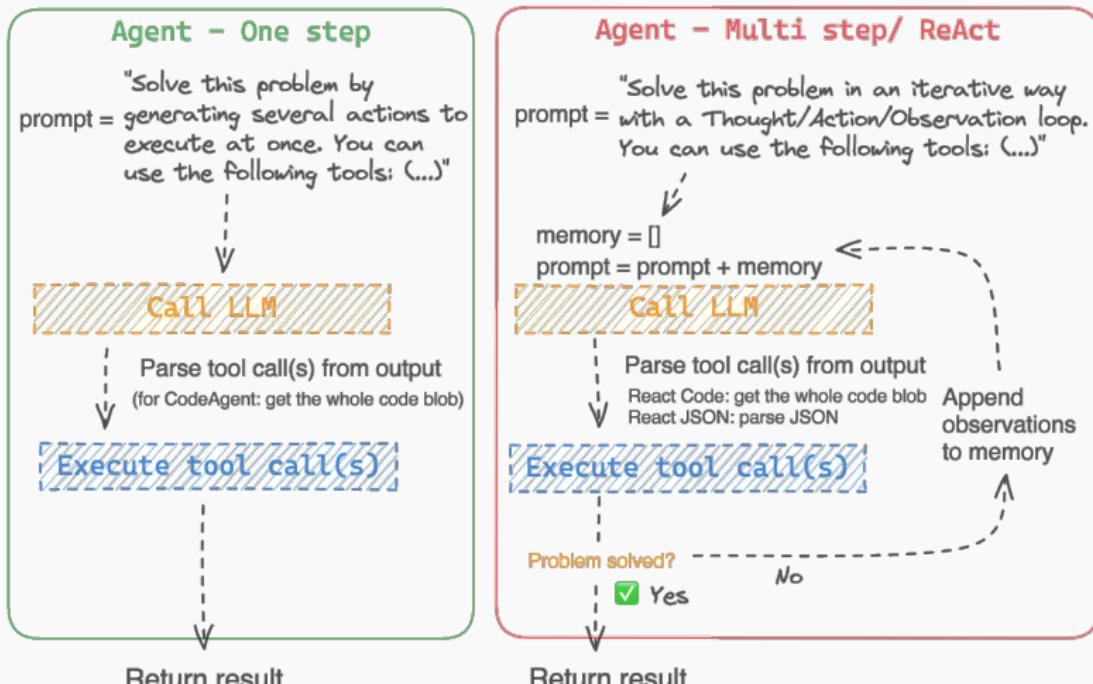
Esto generará:

```
<tool_call>  
{"arguments": {"location": "Paris, France", "unit": "celsius"}, "name":  
    "get_current_temperature"}  
</tool_call><|im_end|>
```

El modelo sabe llamar a la función correcta para resolver el problema.

Agente de Lenguaje en práctica con ReAct y HuggingFace

ReAct es un enfoque de prompts para construir agentes siguiendo ciclos de Pensamiento \Rightarrow Acción \Rightarrow Observación hasta resolver la tarea [65].



Agente de Lenguaje en práctica con ReAct y HuggingFace

ReAct es un enfoque de prompts para construir agentes siguiendo ciclos de **Pensamiento ⇒ Acción ⇒ Observación** hasta resolver la tarea [65].

- El agente tiene acceso a diversas herramientas ya codificadas o que puedes crear usando `transformers.agents.Tool`.
- Es posible acceder a sus "*pensamientos*" que son explicaciones sobre su comportamiento mientras planifica.
- Es posible acceder a los estados internos:
 - sus acciones,
 - sus "*pensamientos*": explicaciones de su comportamiento durante la planificación.

Puedes ver el system prompt de la clase `ReActAgent` aquí. También revisa el blogpost y el tutorial.

Ejemplo en video



Figure 20: Un video que ilustra a un agente respondiendo a una consulta

¿Pueden los LLMs simular interacciones sociales?

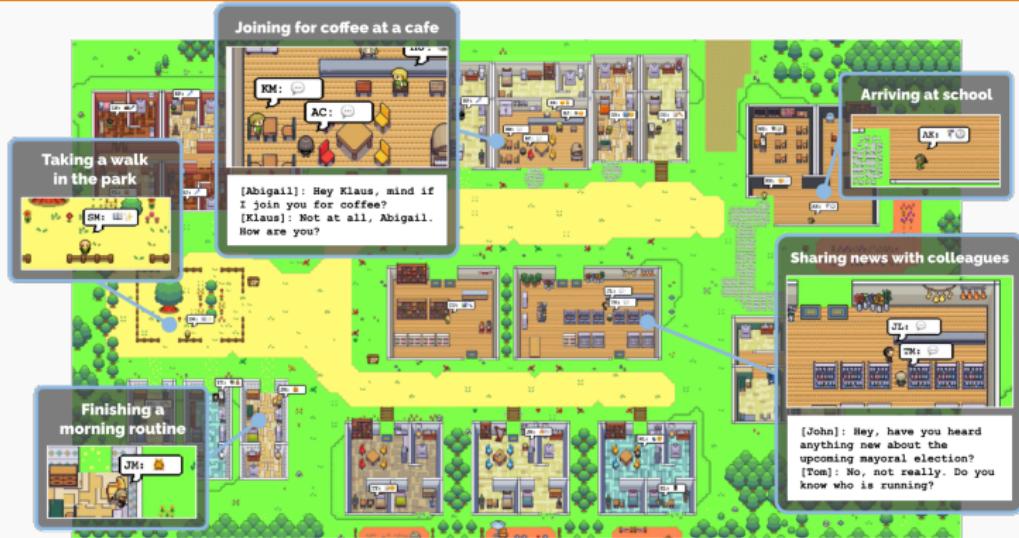


Figure 21: Generative Agents: Interactive Simulacra of Human Behavior [34]

¿Pueden los LLMs simular interacciones sociales?

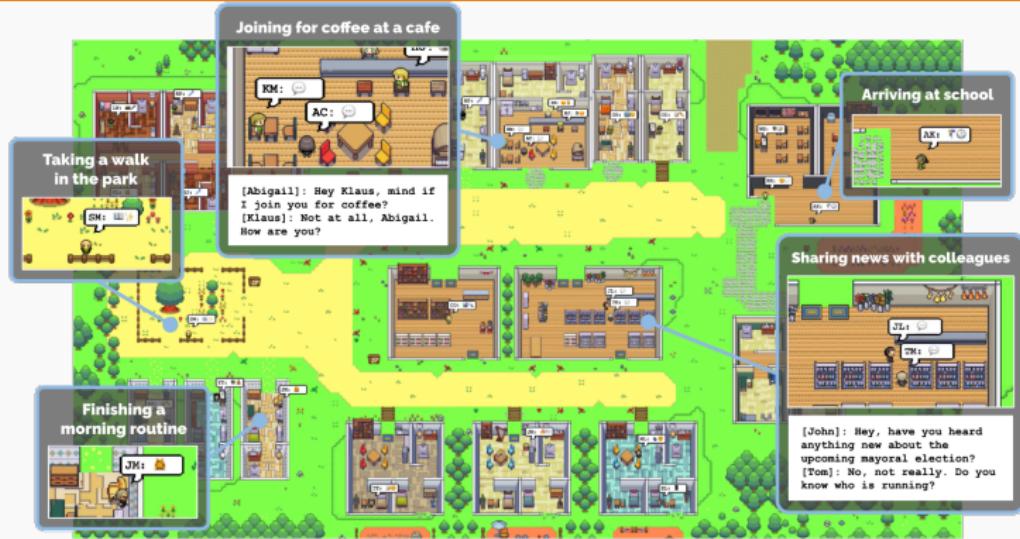


Figure 21: Generative Agents: Interactive Simulacra of Human Behavior [34]

No es tan simple! [69]

En realidad, los LLMs tienen dificultades para acceder de forma humana al estado mental de su interlocutor (objetivos, creencias, deseos, conocimiento, ...).

Algunos recursos externos

- Modelos de Lenguaje a Gran Escala:
 - El curso [Large Language Model](#) de Percy Liang (Stanford).
 - Diapositivas [LLM](#) de Felipe Bravo.
- Agentes:
 - Clase completa sobre [Agentes LLM](#) por Dawn Song (Berkeley).
 - Tutorial muy completo [aquí](#).
 - Tutorial [Agent 101](#) de HuggingFace.
 - Un [post en el blog](#) sobre cómo alcanzaron buen desempeño en el benchmark GAIA.

Questions?

References i

-  G. Bai, J. Liu, X. Bu, Y. He, J. Liu, Z. Zhou, Z. Lin, W. Su, T. Ge, B. Zheng, and W. Ouyang.
MT-Bench-101: A Fine-Grained Benchmark for Evaluating Large Language Models in Multi-Turn Dialogues.
In ACL, 2024.
-  Y. Bai, C. L. Apr, A. Askell, A. Chen, N. Dassarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. El-showk, N. Elhage, Z. Hatfield-dodds, D. Hernandez, T. Hume, S. Johnston, S. Kravec, L. Lovitt, N. Nanda, C. Olsson, D. Amodei, T. Brown, J. Clark, S. Mccandlish, C. Olah, B. Mann, and J. Kaplan.
Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback.

References ii

- 
- Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Chen, C. Olsson, C. Olah, D. Hernandez, D. Drain, D. Ganguli, D. Li, E. Tran-Johnson, E. Perez, J. Kerr, J. Mueller, J. Ladish, J. Landau, K. Ndousse, K. Lukosuite, L. Lovitt, M. Sellitto, N. Elhage, N. Schiefer, N. Mercado, N. DasSarma, R. Lasenby, R. Larson, S. Ringer, S. Johnston, S. Kravec, S. E. Showk, S. Fort, T. Lanham, T. Telleen-Lawton, T. Conerly, T. Henighan, T. Hume, S. R. Bowman, Z. Hatfield-Dodds, B. Mann, D. Amodei, N. Joseph, S. McCandlish, T. Brown, and J. Kaplan.

Constitutional AI: Harmlessness from AI Feedback.

2022.

References iii

-  A. Barbaresi.
Trafilatura: A web scraping library and command-line tool for text discovery and extraction.
ACL-IJCNLP 2021 - 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Proceedings of the System Demonstrations, pages 122–131, 2021.
-  T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei.
Language Models are Few-Shot Learners.
2020.

-  T. Cai, X. Wang, T. Ma, X. Chen, and D. Zhou.
Large Language Models As Tool Makers.
12th International Conference on Learning Representations, ICLR
2024, pages 1–23, 2024.
-  W. L. Chiang, L. Zheng, Y. Sheng, A. N. Angelopoulos, T. Li, D. Li,
B. Zhu, H. Zhang, M. I. Jordan, J. E. Gonzalez, and I. Stoica.
Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference.
Proceedings of the 41 st International Conference on Machine Learning, 235:8359–8388, 2024.

References v

- 📄 E. Choi, H. He, M. Iyyer, M. Yatskar, W. T. Yih, Y. Choi, P. Liang, and L. Zettlemoyer.

QUAC: Question answering in context.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018, pages 2174–2184, 2018.

- 📄 H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Robert, D. Zhou, Q. V. Le, and J. Wei.

Scaling Instruction-Finetuned Language Models.

Journal of Machine Learning Research, 25:1–53, 2024.

References vi

-  M. Conover, M. Hayes, A. Mathur, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, and R. Xin.
Free Dolly: Introducing the World's First Truly Open Instruction-Tuned LLM, 2023.
-  J. Dodge, M. Sap, A. Marasović, W. Agnew, G. Ilharco, D. Groeneveld, M. Mitchell, and M. Gardner.
Documenting Large Webtext Corpora: A Case Study on the Colossal Clean Crawled Corpus.
EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings, (Table 1):1286–1305, 2021.
-  Y. Dubois, B. Galambosi, P. Liang, and T. B. Hashimoto.
Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators.
pages 1–11, 2024.

-  L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima, S. Presser, and C. Leahy. **The Pile: An 800GB Dataset of Diverse Text for Language Modeling.** 2020.
-  S. Gunasekar, Y. Zhang, J. Aneja, C. Cesar, T. Mendes, A. D. Giorno, S. Gopi, M. Javaheripi, P. Kauffmann, G. de Rosa, O. Saarikivi, A. Salim, S. Shah, and Y. Singh B. **Textbooks Are All You Need.** pages 1–26, 2023.
-  D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. **Measuring Massive Multitask Language Understanding.** In ICLR, 2021.

-  D. Hernandez, T. Brown, T. Conerly, N. DasSarma, D. Drain, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, T. Henighan, T. Hume, S. Johnston, B. Mann, C. Olah, C. Olsson, D. Amodei, N. Joseph, J. Kaplan, and S. McCandlish.
Scaling Laws and Interpretability of Learning from Repeated Data.
pages 1–23, 2022.
-  Y. Hwang, Y. Kim, H. Bae, H. Lee, J. Bang, and K. Jung.
Dialogizer: Context-aware Conversational-QA Dataset Generation from Textual Sources.
EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings, pages 8806–8828, 2023.

-  S. Jeong, J. Baek, S. J. Hwang, and J. C. Park.
Realistic Conversational Question Answering with Answer Selection based on Calibrated Confidence and Uncertainty Measurement.
EACL 2023 - 17th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, pages 477–490, 2023.
-  M. Joshi, E. Choi, D. S. Weld, and L. Zettlemoyer.
TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension.
ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), 1:1601–1611, 2017.

References x

-  D. Kocetkov, R. Li, L. B. Allal, J. Li, C. Mou, C. M. Ferrandis, Y. Jernite, M. Mitchell, S. Hughes, T. Wolf, D. Bahdanau, L. von Werra, and H. de Vries.
The Stack: 3 TB of permissively licensed source code.
Transactions on Machine Learning Research, pages 1–27, 2022.
-  T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa.
Large Language Models are Zero-Shot Reasoners.
Number NeurIPS, 2022.
-  LAION.
The Open-Instruction-Generalist Dataset, 2023.

-  K. Lee, D. Ippolito, A. Nystrom, C. Zhang, D. Eck, C. Callison-Burch, and N. Carlini.
Deduplicating Training Data Makes Language Models Better.
Proceedings of the Annual Meeting of the Association for Computational Linguistics, 1:8424–8445, 2022.
-  P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela.
Retrieval-augmented generation for knowledge-intensive NLP tasks.
Advances in Neural Information Processing Systems, 2020-Decem, 2020.

-  B. Li, T. Yan, Y. Pan, Z. Xu, J. Luo, R. Ji, S. Liu, H. Dong, Z. Lin, and Y. Wang.
MMedAgent: Learning to Use Medical Tools with Multi-modal Agent.
In EMNLP, 2024.
-  X. Li, T. Zhang, Y. Dubois, R. Taori, I. Gulrajani, C. Guestrin, P. Liang, and T. B. Hashimoto.
AlpacaEval: An Automatic Evaluator of Instruction-following Models.
\url{https://github.com/tatsu-lab/alpaca_eval}, 2023.
-  Z. Li, X. Xu, T. Shen, C. Xu, J.-C. Gu, and C. Tao.
Leveraging Large Language Models for NLG Evaluation: A Survey.
In EMNLP, pages 16028–16045, 2024.

-  S. Lin, J. Hilton, and O. Evans.
TruthfulQA: Measuring How Models Mimic Human Falsehoods.
Proceedings of the Annual Meeting of the Association for Computational Linguistics, 1:3214–3252, 2022.

-  S. Longpre, L. Hou, T. Vu, A. Webson, H. W. Chung, Y. Tay, D. Zhou, Q. V. Le, B. Zoph, J. Wei, and A. Roberts.
The Flan Collection: Designing Data and Methods for Effective Instruction Tuning.
In Proceedings of Machine Learning Research, volume 202, pages 22631–22648, 2023.

-  J. Lu, T. Holleis, Y. Zhang, B. Aumayer, F. Nan, F. Bai, S. Ma, S. Ma, M. Li, G. Yin, Z. Wang, and R. Pang.
ToolSandbox: A Stateful, Conversational, Interactive Evaluation Benchmark for LLM Tool Use Capabilities.
2024.
-  G. Mialon, C. Fourrier, C. Swift, T. Wolf, Y. LeCun, and T. Scialom.
Gaia: a Benchmark for General Ai Assistants.
12th International Conference on Learning Representations, ICLR
2024, pages 1–24, 2024.

-  A. Mohammadshahi, T. Scialom, M. Yazdani, P. Yanki, A. Fan, J. Henderson, and M. Saeidi.

RQUGE: Reference-Free Metric for Evaluating Question Generation by Answering the Question.

Proceedings of the Annual Meeting of the Association for Computational Linguistics, pages 6845–6867, 2023.

-  L. Ouyang, J. Wu, X. Jiang, D. Ameida, C. L. Wainwright, P. Mishkin, C. L. Mar, J. Hilton, A. Askell, P. Christiano, J. Leike, and R. Lowe.

Training language models to follow instructions with human feedback.

arXiv, <https://op>, 2022.

References xvi

-  J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein.
Generative Agents: Interactive Simulacra of Human Behavior.
In UIST '23: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, volume 1. Association for Computing Machinery, 2023.
-  G. Penedo, H. Kydlíček, L. B. Allal, A. Lozhkov, M. Mitchell, C. Raffel, L. Von Werra, and T. Wolf.
The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale.
2024.

References xvii

-  G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, and J. Launay.
The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only.
2023.
-  J.-L. Peng, S. Cheng, E. Diau, Y.-Y. Shih, P.-H. Chen, Y.-T. Lin, and Y.-N. Chen.
A Survey of Useful LLM Evaluation.
2024.
-  O. Press, M. Zhang, S. Min, L. Schmidt, N. A. Smith, and M. Lewis.
Measuring and Narrowing the Compositionality Gap in Language Models.
2022.

References xviii

-  Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y. Lu, Y. Lin, X. Cong, X. Tang, B. Qian, S. Zhao, L. Hong, R. Tian, R. Xie, J. Zhou, M. Gerstein, D. Li, Z. Liu, and M. Sun.
Toolllm: Facilitating Large Language Models To Master 16000+ Real-World Apis.
12th International Conference on Learning Representations, ICLR 2024, 2024.
-  R. Rafailov, A. Sharma, E. Mitchell, S. Ermon, C. D. Manning, and C. Finn.
Direct Preference Optimization: Your Language Model is Secretly a Reward Model.
2023.

-  S. Reddy, D. Chen, and C. D. Manning.
CoQA: A Conversational Question Answering Challenge.
Transactions of the Association for Computational Linguistics,
7:249–266, 2019.
-  V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai,
A. Chaffin, A. Stiegler, T. L. Scao, A. Raja, M. Dey, M. S. Bari,
C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim,
G. Chhablani, N. Nayak, D. Datta, J. Chang, M. T.-J. Jiang,
H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden,
T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Fevry, J. A.
Fries, R. Teehan, S. Biderman, L. Gao, T. Bers, T. Wolf, and A. M.
Rush.
**Multitask Prompted Training Enables Zero-Shot Task
Generalization.**
2021.

References xx

-  R. Schaeffer, B. Miranda, and S. Koyejo.
Are Emergent Abilities of Large Language Models a Mirage?
In Neurips, pages 1–14, 2023.
-  T. Schick, M. Lomeli, J. Dwivedi-yu, and R. Dessì.
Toolformer: Language Models Can Teach Themselves to Use Tools.
Neurips, 2023.
-  R. Sennrich, B. Haddow, and A. Birch.
Neural Machine Translation of Rare Words with Subword Units.
In ACL, pages 1715–1725, 2016.

References xxi

-  V. Sharma, K. Padthe, N. Ardalani, K. Tirumala, R. Howes, H. Xu, P.-Y. Huang, S.-W. Li, A. Aghajanyan, G. Ghosh, and L. Zettlemoyer.
Text Quality-Based Pruning for Efficient Training of Language Models.
2024.
-  Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang.
HuggingGPT: Solving AI Tasks with ChatGPT and its Friends in Hugging Face.
Advances in Neural Information Processing Systems,
36(NeurIPS):1–27, 2023.

References xxii

-  N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao.
Reflexion: Language Agents with Verbal Reinforcement Learning.
Advances in Neural Information Processing Systems, 36(NeurIPS),
2023.
-  S. Singh, F. Vargus, D. Dsouza, B. F. Karlsson, A. Mahendiran,
W.-y. Ko, H. Shandilya, J. Patel, D. Mataciunas, L. OMahony,
M. Zhang, R. Hettiarachchi, J. Wilson, M. Machado, L. S. Moura,
D. Krzemiński, H. Fadaei, I. Ergün, I. Okoh, A. Alaagib,
O. Mudannayake, Z. Alyafeai, V. M. Chien, S. Ruder, S. Guthikonda,
E. A. Alghamdi, S. Gehrmann, N. Muennighoff, M. Bartolo,
J. Kreutzer, A. Üstün, M. Fadaee, and S. Hooker.
Aya Dataset: An Open-Access Collection for Multilingual Instruction Tuning.
In ACL, 2024.

-  L. Soldaini, R. Kinney, A. Bhagia, D. Schwenk, D. Atkinson, R. Authur, B. Bogin, K. Chandu, J. Dumas, Y. Elazar, V. Hofmann, A. H. Jha, S. Kumar, L. Lucy, X. Lyu, N. Lambert, I. Magnusson, J. Morrison, N. Muennighoff, A. Naik, C. Nam, M. E. Peters, A. Ravichander, K. Richardson, Z. Shen, E. Strubell, N. Subramani, O. Tafjord, P. Walsh, L. Zettlemoyer, N. A. Smith, H. Hajishirzi, I. Beltagy, D. Groeneveld, J. Dodge, and K. Lo.

Dolma: an Open Corpus of Three Trillion Tokens for Language Model Pretraining Research.

In ACL, 2024.

References xxiv

- 
- A. Srivastava, A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, A. Kluska, A. Lewkowycz, A. Agarwal, A. Power, A. Ray, A. Warstadt, A. W. Kocurek, A. Safaya, A. Tazarv, A. Xiang, A. Parrish, A. Nie, A. Hussain, A. Askell, A. Dsouza, A. Slone, A. Rahane, A. S. Iyer, A. Andreassen, A. Madotto, A. Santilli, A. Stuhlmüller, A. Dai, A. La, A. Lampinen, A. Zou, A. Jiang, A. Chen, A. Vuong, A. Gupta, A. Gottardi, A. Norelli, A. Venkatesh, A. Gholamidavoodi, A. Tabassum, A. Menezes, A. Kirubarajan, A. Mullokandov, A. Sabharwal, A. Herrick, A. Efrat, A. Erdem, A. Karakaş, B. R. Roberts, B. S. Loe, B. Zoph, B. Bojanowski, B. Özyurt, B. Hedayatnia, B. Neyshabur, B. Inden, B. Stein, B. Ekmekci, B. Y. Lin, B. Howald, B. Orinion, C. Diao, C. Dour, C. Stinson, C. Argueta, C. F. Ramírez, C. Singh, C. Rathkopf, C. Meng, C. Baral, C. Wu, C. Callison-Burch, C. Waites, C. Voigt,

References xxv

- C. D. Manning, C. Potts, C. Ramirez, C. E. Rivera, C. Siro,
C. Raffel, C. Ashcraft, C. Garbacea, D. Sileo, D. Garrette,
D. Hendrycks, D. Kilman, D. Roth, D. Freeman, D. Khashabi,
D. Levy, D. M. González, D. Perszyk, D. Hernandez, D. Chen,
D. Ippolito, D. Gilboa, D. Dohan, D. Drakard, D. Jurgens, D. Datta,
D. Ganguli, D. Emelin, D. Kleyko, D. Yuret, D. Chen, D. Tam,
D. Hupkes, D. Misra, D. Buzan, D. C. Mollo, D. Yang, D.-H. Lee,
D. Schrader, E. Shutova, E. D. Cubuk, E. Segal, E. Hagerman,
E. Barnes, E. Donoway, E. Pavlick, E. Rodola, E. Lam, E. Chu,
E. Tang, E. Erdem, E. Chang, E. A. Chi, E. Dyer, E. Jerzak, E. Kim,
E. E. Manyasi, E. Zheltonozhskii, F. Xia, F. Siar,
F. Martínez-Plumed, F. Happé, F. Chollet, F. Rong, G. Mishra, G. I.
Winata, G. de Melo, G. Kruszewski, G. Parascandolo, G. Mariani,
G. Wang, G. Jaimovich-López, G. Betz, G. Gur-Ari, H. Galijasevic,
H. Kim, H. Rashkin, H. Hajishirzi, H. Mehta, H. Bogar, H. Shevlin,

References xxvi

H. Schütze, H. Yakura, H. Zhang, H. M. Wong, I. Ng, I. Noble,
J. Jumelet, J. Geissinger, J. Kernion, J. Hilton, J. Lee, J. F. Fisac,
J. B. Simon, J. Koppel, J. Zheng, J. Zou, J. Kocoń, J. Thompson,
J. Wingfield, J. Kaplan, J. Radom, J. Sohl-Dickstein, J. Phang,
J. Wei, J. Yosinski, J. Novikova, J. Bosscher, J. Marsh, J. Kim,
J. Taal, J. Engel, J. Alabi, J. Xu, J. Song, J. Tang, J. Waweru,
J. Burden, J. Miller, J. U. Balis, J. Batchelder, J. Berant,
J. Frohberg, J. Rozen, J. Hernandez-Orallo, J. Boudeman, J. Guerr,
J. Jones, J. B. Tenenbaum, J. S. Rule, J. Chua, K. Kanclerz,
K. Livescu, K. Krauth, K. Gopalakrishnan, K. Ignatyeva, K. Markert,
K. D. Dhole, K. Gimpel, K. Omondi, K. Mathewson, K. Chiaffullo,
K. Shkaruta, K. Shridhar, K. McDonell, K. Richardson, L. Reynolds,
L. Gao, L. Zhang, L. Dugan, L. Qin, L. Contreras-Ochando, L.-P.
Morency, L. Moschella, L. Lam, L. Noble, L. Schmidt, L. He, L. O.
Colón, L. Metz, L. K. Senel, M. Bosma, M. Sap, M. ter Hoeve,

References xxvii

M. Farooqi, M. Faruqui, M. Mazeika, M. Baturan, M. Marelli,
M. Maru, M. J. R. Quintana, M. Tolkiehn, M. Giulianelli, M. Lewis,
M. Potthast, M. L. Leavitt, M. Hagen, M. Schubert, M. O.
Baitemirova, M. Arnaud, M. McElrath, M. A. Yee, M. Cohen,
M. Gu, M. Ivanitskiy, M. Starritt, M. Strube, M. Swędrowski,
M. Bevilacqua, M. Yasunaga, M. Kale, M. Cain, M. Xu, M. Suzgun,
M. Walker, M. Tiwari, M. Bansal, M. Aminnaseri, M. Geva,
M. Gheini, M. V. T, N. Peng, N. A. Chi, N. Lee, N. G.-A. Krakover,
N. Cameron, N. Roberts, N. Doiron, N. Martinez, N. Nangia,
N. Deckers, N. Muennighoff, N. S. Keskar, N. S. Iyer, N. Constant,
N. Fiedel, N. Wen, O. Zhang, O. Agha, O. Elbaghdadi, O. Levy,
O. Evans, P. A. M. Casares, P. Doshi, P. Fung, P. P. Liang, P. Vicol,
P. Alipoormolabashi, P. Liao, P. Liang, P. Chang, P. Eckersley, P. M.
Htut, P. Hwang, P. Miłkowski, P. Patil, P. Pezeshkpour, P. Oli,
Q. Mei, Q. Lyu, Q. Chen, R. Banjade, R. E. Rudolph, R. Gabriel,

References xxviii

- R. Habacker, R. Risco, R. Millière, R. Garg, R. Barnes, R. A. Saurous, R. Arakawa, R. Raymaekers, R. Frank, R. Sikand, R. Novak, R. Sitelew, R. LeBras, R. Liu, R. Jacobs, R. Zhang, R. Salakhutdinov, R. Chi, R. Lee, R. Stovall, R. Teehan, R. Yang, S. Singh, S. M. Mohammad, S. Anand, S. Dillavou, S. Shleifer, S. Wiseman, S. Gruetter, S. R. Bowman, S. S. Schoenholz, S. Han, S. Kwatra, S. A. Rous, S. Ghazarian, S. Ghosh, S. Casey, S. Bischoff, S. Gehrman, S. Schuster, S. Sadeghi, S. Hamdan, S. Zhou, S. Srivastava, S. Shi, S. Singh, S. Asaadi, S. S. Gu, S. Pachchigar, S. Toshniwal, S. Upadhyay, Shyamolima, Debnath, S. Shakeri, S. Thormeyer, S. Melzi, S. Reddy, S. P. Makini, S.-H. Lee, S. Torene, S. Hatwar, S. Dehaene, S. Divic, S. Ermon, S. Biderman, S. Lin, S. Prasad, S. T. Piantadosi, S. M. Shieber, S. Misherghi, S. Kiritchenko, S. Mishra, T. Linzen, T. Schuster, T. Li, T. Yu, T. Ali, T. Hashimoto, T.-L. Wu, T. Desbordes,

References xxix

T. Rothschild, T. Phan, T. Wang, T. Nkinyili, T. Schick, T. Kornev,
T. Tunduny, T. Gerstenberg, T. Chang, T. Neeraj, T. Khot,
T. Shultz, U. Shaham, V. Misra, V. Demberg, V. Nyamai,
V. Raunak, V. Ramasesh, V. U. Prabhu, V. Padmakumar,
V. Sri Kumar, W. Fedus, W. Saunders, W. Zhang, W. Vossen,
X. Ren, X. Tong, X. Zhao, X. Wu, X. Shen, Y. Yaghoobzadeh,
Y. Lakretz, Y. Song, Y. Bahri, Y. Choi, Y. Yang, Y. Hao, Y. Chen,
Y. Belinkov, Y. Hou, Y. Hou, Y. Bai, Z. Seid, Z. Zhao, Z. Wang,
Z. J. Wang, Z. Wang, and Z. Wu.

**Beyond the Imitation Game: Quantifying and extrapolating
the capabilities of language models.**

2022.

References xxx

-  R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto.
Stanford Alpaca: An Instruction-following LLaMA model.
\url{https://github.com/tatsu-lab/stanford_alpaca}, 2023.
-  K. Tirumala, D. Simig, A. Aghajanyan, and A. S. Morcos.
D4: Improving LLM Pretraining via Document De-Duplication and Diversification.
(NeurIPS), 2023.
-  X. Wang, Z. Hu, P. Lu, Y. Zhu, and J. Zhang.
SCIBENCH : Evaluating College-Level Scientific Problem-Solving Abilities of Large Language Models.
In MATH-AI: The 3rd Workshop on Mathematical Reasoning and AI
© Neurips, number NeurIPS, 2023.

-  X. Wang and D. Zhou.
Chain-of-Thought Reasoning Without Prompting.
pages 1–23, 2024.
-  Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, W. Ren, A. Arulraj, X. He, Z. Jiang, T. Li, M. Ku, K. Wang, A. Zhuang, R. Fan, X. Yue, and W. Chen.
MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark.
In NeurIPS, pages 1–24, 2024.

-  M. Weber, D. Y. Fu, Q. Anthony, Y. Oren, S. Adams, A. Alexandrov, X. Lyu, H. Nguyen, X. Yao, V. Adams, B. Athiwaratkun, R. Chalamala, K. Chen, M. Ryabinin, T. Dao, P. Liang, C. Ré, I. Rish, and C. Zhang.
RedPajama: an Open Dataset for Training Large Language Models.
In NeurIPS, pages 1–31, 2024.
-  J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le.
Finetuned Language Models Are Zero-Shot Learners.
ICLR, pages 1–41, 2022.

References xxxiii

-  J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus.
Emergent Abilities of Large Language Models.
Transactions on Machine Learning Research, pages 1–30, 2022.
-  J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou.
Chain-of-Thought Prompting Elicits Reasoning in Large Language Models.
(NeurIPS):1–43, 2022.
-  T. Wu, J. Lan, W. Yuan, J. Jiao, J. Weston, and S. Sukhbaatar.
Thinking LLMs: General Instruction Following with Thought Generation.
pages 1–28, 2024.

-  C. Yang, X. Wang, Y. Lu, H. Liu, Q. V. Le, D. Zhou, and X. Chen.
Large Language Models as Optimizers.
2023.
-  S. Yao.
Language Agents: From Next-Token Prediction to Digital Automation
PhD thesis, 2024.
-  S. Yao, D. Yu, J. Zhao, and T. L. Griffiths.
Tree of Thoughts : Deliberate Problem Solving with Large Language Models.
In Neurips, number 1, pages 1–11, 2023.

-  S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao.
React: Synergizing Reasoning and Acting in Language Models.
11th International Conference on Learning Representations, ICLR
2023, pages 1–33, 2023.
-  M. J. Zhang, Z. Wang, J. D. Hwang, Y. Dong, O. Delalleau, Y. Choi, E. Choi, X. Ren, and V. Pyatkin.
Diverging Preferences: When do Annotators Disagree and do Models Know?
(Table 1):1–19, 2024.

References xxxvi

-  L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica.
Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena.
(NeurIPS):1–29, 2023.
-  C. Zhou, P. Liu, P. Xu, S. Iyer, J. Sun, Y. Mao, X. Ma, A. Efrat, P. Yu, L. Yu, S. Zhang, G. Ghosh, M. Lewis, L. Zettlemoyer, and O. Levy.
LIMA: Less Is More for Alignment.
pages 1–15, 2023.
-  X. Zhou, Z. Su, T. Eisape, H. Kim, and M. Sap.
Is this the real life? Is this just fantasy? The Misleading Success of Simulating Social Interactions With LLMs.
In EMNLP, 2024.

-  Y. Zhuang, Y. Yu, K. Wang, H. Sun, and C. Zhang.
ToolQA: A Dataset for LLM Question Answering with External Tools.
Neurips, 2023.