

- Chat basado en un lenguaje de inteligencia artificial
 - Es desarrollado por OpenAI, empresa fundada por Elon Musk en 2015. Tiene más de 175 mil millones de parámetros y 12 capas. GPT-4 tendrá 100 trillones de parámetros, casi 600 veces más que su predecesor y podría lanzarse en cualquier momento de 2023.
 - Su aprendizaje comenzó con una base de datos con 19 millones de entradas. Chat GPT ha sido entrenado con una gran cantidad de datos de texto para poder realizar una amplia variedad de tareas relacionadas con el lenguaje natural. Estos datos de texto incluyen libros, artículos, noticias, conversaciones, entre otros, que se utilizan para enseñarle al modelo cómo comprender y generar texto de manera coherente y natural. Por lo tanto, Chat GPT obtiene la información necesaria para generar respuestas complejas a partir de estos datos de texto, que le permiten comprender el contexto y la intención detrás de las preguntas o consultas de los usuarios
 - Este chat está basado en la tecnología GPT o Generative Pre-trained Transformer, un modelo de lenguaje que utiliza el deep learning para generar texto de una forma muy parecida a cómo lo haría un humano. Esta inteligencia artificial aprende a base de texto, por lo que gracias a este entrenamiento conversacional, es capaz de ser cada vez más detallado en sus respuestas y hacerlo de una forma natural.
 - No solo puedes conversar con esta IA, sino que es capaz de redactar con gran precisión, explicar conceptos complejos de una forma clara y concisa, así como generar código programático o resolver operaciones matemáticas.
 - Actualmente para usarlo es totalmente gratuito, pero pronto será de pago. Va a haber una cuenta PRO, el plan “profesional”, el cual ofrecerá beneficios como respuestas más rápidas y una disponibilidad inmediata. ChatGPT, no siempre está disponible, algo que no siempre es posible ella debido a la inmensa demanda que existe por ella.
 - Las peticiones pueden ser tanto en español como en inglés.
 - Además, el modelo también puede utilizar otros tipos de información, como imágenes o vídeos, para mejorar su capacidad para comprender el mundo que lo rodea y generar respuestas más precisas y coherentes.
 - trabaja en inglés de manera nativa. Las preguntas que se le hacen en español, o cualquier otro idioma, las traduce al inglés, procesa, genera la contestación y la vuelve a traducir al idioma en el que se le ha hecho la pregunta. Esto puede ser menos exacto que si se le pregunta directamente en inglés. Las entradas de entrenamiento están en inglés.
 - Modelos de lenguaje con los que trabaja: ada, babbage, curie, davinci. Davinci es el más avanzado y es con el que hemos trabajado.
 - Microsoft será proveedor exclusivo de servicios en la nube para OpenAI a través de Azure.
- Microsoft ampliará su infraestructura de supercomputadoras potenciadas por Azure, para que OpenAI continúe entrenando a sus modelos de inteligencia artificial.
- La intención de Microsoft es que tanto usuarios finales como empresariales puedan incorporar dichas herramientas en sus propias aplicaciones. ¿Y cómo planea lograrlo? A través de Azure, por supuesto.
 - Microsoft trabaja para implementar ChatGPT en Bing, para por fin competir cara a cara con Google, como así también en las principales aplicaciones de Office, como Word y PowerPoint.
 - Tras una inversión billonaria, Microsoft quedaría como dueño del 49% de la compañía dirigida por Sam Altman. Otro 49% pertenecería a los demás inversores (Elon Musk y Sam Altman), mientras que el 2% restante quedaría en manos de OpenAI Inc., la casa matriz sin fines de lucro que maneja el laboratorio de IA.

Versiones

- **GPT-1 (2018):** Este modelo podía ser re-entrenado con relativamente pocos datos, y alcanzar resultados SotA (State of the Art) en múltiples benchmarks.
- **GPT-2 (2019):** 10X parámetros con respecto al anterior, por lo que re entrenar el modelo pasaba a ser una tarea compleja por la infraestructura que requería. También, cambió la data con la que estaba entrenado, es decir “WebText” (data de la web, por ejemplo, Reddit). Capacidad de generar texto “coherente”. Capacidad de

hacer “few-shot learning”, es decir, aprender sin re-entrenar el modelo, usando pocos ejemplos en tareas que el modelo nunca vio en su entrenamiento inicial. A nivel aplicativo, GPT-2 podía generar títulos de noticias muy realistas. También fue adoptado para generar imágenes (feedback loop entre NLP y Computer Vision), sin embargo, cuando intentó ser adaptado para crear herramientas conversacionales, las mismas fallaron.

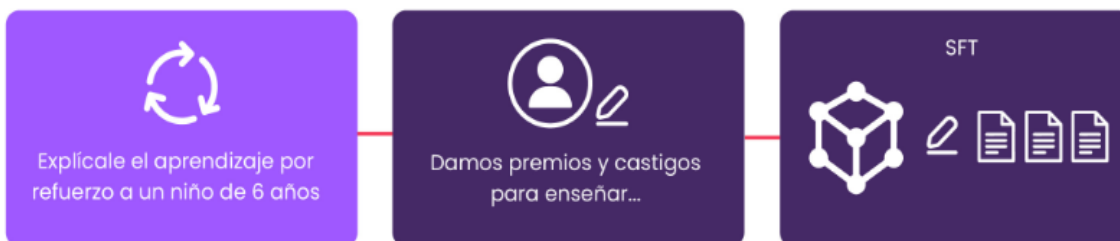
- **GPT-3 (2020):** cambió (otra vez) la cantidad de parámetros del modelo (de 1.5 billones a 175 billones), y también cambió el dataset. Agregaron muchísima más data de la Web, como por ejemplo, Common Crawl, Wikipedia y WebText actualizado. Al igual que su predecesor, estableció nuevos SotA en múltiples puntos de referencia y mejoró considerablemente su capacidad de “zero-shot learning”. Mostró parcialidad (religiosos, de género, etc).

Proceso de entrenamiento de ChatGPT

1º Recopila datos de demostración y entrena con datos supervisados.

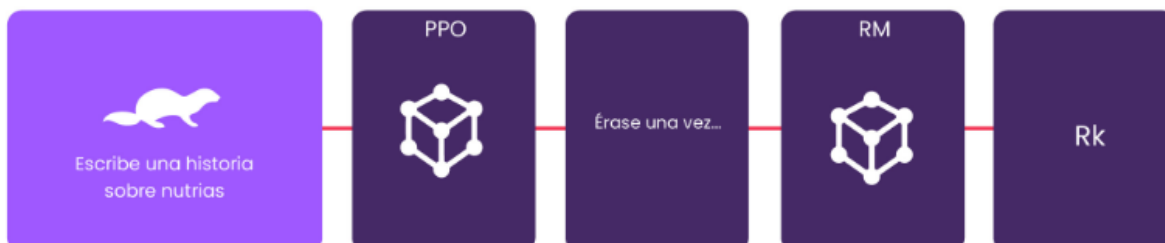
2º El segundo paso es el más interesante, ya que se le propone al modelo diferentes respuestas y lo que el feedback humano proporciona es un ranking de respuestas de más deseable a menos. Usando esta información, se puede entrenar un modelo que es “recompensado” (Reward Model) y que capture las preferencias humanas, de forma de así poder reducir la falta de alineamiento.

Se prueba un prompt de nuestro conjunto de prompts Un etiquetador demuestra el comportamiento de salida deseado Estos datos se utilizan para afinar GPT-3.5 con aprendizaje supervisado.



3º Optimiza una policy (término de Reinforcement Learning) contra el Reward Model utilizando el algoritmo de aprendizaje por refuerzo de PPO.

Se muestrea un nuevo prompt del conjunto de datos. El modelo PPO se inicializa desde la policy supervisada. La política genera una respuesta. El modelo de recompensa calcula una recompensa por la salida. La recompensa se utiliza para actualizar la policy mediante PPO.



MODEL	TRAINING	USAGE
Ada	\$0.0004 / 1K tokens	\$0.0016 / 1K tokens
Babbage	\$0.0006 / 1K tokens	\$0.0024 / 1K tokens
Curie	\$0.0030 / 1K tokens	\$0.0120 / 1K tokens
Davinci	\$0.0300 / 1K tokens	\$0.1200 / 1K tokens

tokens: trozos de palabras. Hay un tokenizador en la web de openai para poder prever cuántos tokens tiene nuestra consulta y así sacar el precio del servicio. Cobran, porque los cálculos predictivos se hacen usando GPU, osea, gran consumo de luz eléctrica.

Hay una aplicación en la web <https://gpttools.com/comparisontool> que compara cual de los 4 modelos predictivos es mejor dependiendo de lo que se introduzca

ABC de Sevilla, periódico digital líder en Andalucía y Sevilla, ofrece noticias en español sobre toda nuestra región. Toda la información y servicios.

Output

Elapsed time: **4632 milliseconds**

Engine:

Temperature:

Top P:

Tokens:

Stop:

LATEST MODEL	DESCRIPTION	MAX REQUEST	TRAINING DATA
text-davinci-003	El modelo GPT-3 más capaz. Puede realizar cualquier tarea que puedan realizar los otros modelos, a menudo con mayor calidad, mayor duración y mejor seguimiento de instrucciones. También admite la inserción de finalizaciones dentro del texto.	4,000 tokens	Up to Jun 2021
text-curie-001	Muy capaz, más rápido y de menor costo que Davinci.	2,048 tokens	Up to Oct 2019
text-babbage-001	Capaz de realizar tareas sencillas, muy rápidas y de menor costo.	2,048 tokens	Up to Oct 2019
text-ada-001	Capaz de tareas muy simples, generalmente el modelo más rápido de la serie GPT-3 y el costo más bajo.	2,048 tokens	Up to Oct 2019

MODELOS

Davinci

Davinci es la familia de modelos más potente, puede realizar cualquier tarea que los otros modelos pueden realizar y, a menudo, con menos instrucciones. Para aplicaciones que requieren mucha comprensión del contenido, como resúmenes para una audiencia específica y generación de contenido creativo, Davinci producirá los mejores resultados. Estas mayores capacidades requieren más recursos informáticos, por lo que Davinci cuesta más por llamada a API y no es tan rápido como los otros modelos.

Otra área en la que brilla Davinci es en la comprensión de la intención del texto. Davinci es bastante bueno resolviendo muchos tipos de problemas lógicos y explicando los motivos. Davinci ha podido resolver algunos de los problemas de IA más desafiantes que involucran causa y efecto.

Bueno en: intención compleja, causa y efecto, resúmenes.

Curie

Curie es extremadamente potente y además muy rápido. Si bien Davinci es más potente cuando se trata de analizar texto complicado, Curie es bastante capaz para muchas tareas matizadas como las clasificaciones y resúmenes. Curie también es bastante bueno respondiendo preguntas y realizando preguntas y respuestas.

Fortalezas: traducción de idiomas, clasificación compleja, texto con sentimiento y resúmenes.

Babbage

Babbage puede realizar tareas sencillas como una clasificación. También es bastante capaz cuando se trata de búsqueda semántica y ver en qué medida los documentos coinciden con las consultas de búsqueda. Muy rápido y de bajo coste.

Fortalezas: clasificación moderada, clasificación de búsqueda semántica

Ada

Ada suele ser el modelo más rápido y puede realizar tareas como analizar texto, corregir direcciones y ciertos tipos de tareas de clasificación que no requieren demasiados matices. El desempeño de Ada a menudo se puede mejorar al proporcionar más contexto.

ALGORITMOS

Supervised fine-tuning (SFT). Aplicamos el modelo en datos etiquetados utilizando aprendizaje supervisado. Entrenamos durante 16 epochs, utilizando una disminución de la tasa de aprendizaje del coseno y un residual dropout de 0,2. Hacemos nuestra selección final del modelo SFT en función del score de RM (Reward modeling) en el conjunto de validación. Encontramos que nuestros modelos SFT se sobreajuste (overfitting) en la pérdida de validación después de 1 epoch; sin embargo, encontramos que el entrenamiento de más epochs ayuda tanto al score del RM como a las calificaciones de preferencia humana, a pesar del overfitting.

Reward Model (RM). A partir del modelo SFT con la última capa un embedding eliminada, entrenamos un modelo para recibir un prompt y una respuesta, y generar una recompensa escalar. En este papel nosotros solo usamos 6B RMs, ya que esto ahorra una gran cantidad de cómputo, y descubrimos que el entrenamiento de 175B RM podría ser inestable y por lo tanto era menos adecuado para ser utilizado como la función de valor durante RL(reinforcement learning from human feedback). En Stiennon et al. (2020), el RM se entrena en un dataset de comparaciones entre dos outputs del modelo en el mismo input. Usan una pérdida de entropía cruzada (cross-entropy), con las comparaciones como etiquetas: la diferencia en las recompensas representan las probabilidades logarítmicas de que una respuesta sea preferida a la otra por un etiquetador humano

Reinforcement learning (RL). Aprendizaje por refuerzo (RL). Una vez más, afinamos el Modelo SFT sobre nuestro entorno utilizando PPO (Schulman et al., 2017). El entorno que muestra avisos de cliente aleatorio y espera una respuesta al prompt. Dado el prompt y el response, produce un reward determinado por el modelo de reward y finaliza el episodio. Además, agregamos una penalización per-token de KL desde el modelo SFT en cada token para mitigar la optimización excesiva del reward model. La función de valor se inicia desde el RM. Llamamos a estos modelos “PPO”.

Baseline: Comparamos el rendimiento de nuestros modelos PPOs con nuestros modelos SFT y GPT-3. También comparamos con GPT-3 cuando se le proporciona un prefijo de pocos disparos al prompt a seguir instrucciones modo (solicitud de GPT-3). Este prefijo se antepone a la instrucción especificada por el usuario. Además, comparamos InstructGPT con el fine-tuning de 175B GPT-3 en FLAN y datasets T0, que consisten en una variedad de tareas NLP(Natural Language Processing), combinadas con natural instrucciones de idioma para cada tarea (los datasets difieren de los datasets de PNL incluidos, y el estilo de instrucciones usadas). Los fine-tune en aproximadamente 1 millón de ejemplos respectivamente y elegimos checkpoint que obtiene el mayor score en el conjunto de validación.

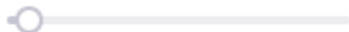
Model

text-davinci-003

Temperature 0.7



Maximum length 256



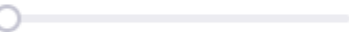
Stop sequences

Enter sequence and press Tab

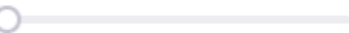
Top P 1



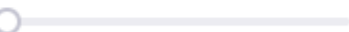
Frequency penalty 0



Presence penalty 0



Best of 1



Inject start text

☒

Inject restart text

☒

Show probabilities

Off

PARÁMETROS:

Temperature: es un valor entre 0 y 1 que esencialmente permite controlar la confianza que debe tener el modelo al hacer estas predicciones. controla la aleatoriedad. La reducción da como resultado finalizaciones menos aleatorias. A medida que la temperatura se acerca a cero, el modelo será determinista y repetitivo. Bajar la temperatura significa que se correrán menos riesgos y las terminaciones serán más precisas. El aumento de la temperatura dará como resultado terminaciones más diversas.

Maximum length: número máximo de tokens a generar.

Stop sequences: hasta cuatro secuencias donde la 'API' dejará de generar más tokens. El texto devuelto no contendrá la secuencia de parada.

Top P: controla la diversidad a través del muestreo del núcleo: significa que se consideran la mitad de todas las opciones ponderadas por probabilidad.

Frequency penalty: cuánto penalizar nuevos tokens en función de su frecuencia existente en el texto hasta el momento. disminuye la probabilidad del modelo de repetir la misma línea palabra por palabra.

Presence penalty: cuánto penalizar nuevos tokens en función de si aparecen en el texto hasta el momento. aumenta la probabilidad de que el modelo hable sobre nuevos temas.

Best of: genera múltiples terminaciones del lado del servidor y muestra solo lo mejor. la transmisión sólo funciona cuando se establece en 1. Como actúa como un multiplicador en la cantidad de finalizaciones, este parámetro puede consumir su cuota de token muy rápidamente.

Inject start text: texto para agregar después del input del usuario para dar formato al modelo para la respuesta.

Inject restart text: texto para agregar después de la generación del modelo para continuar con la estructura modelada.

Show probabilities: alternar el resaltado de 'token' indica la probabilidad de que se genere un token. ayuda a depurar una generación dada o ver opciones alternativas para un token

Resultados a favor:

- Produce escritos de alta calidad, esto puede ayudar a producir contenido más inteligente y convincente.
- Soporta instrucciones más complejas, por lo que puede realizar tareas de razonamiento más avanzadas.
- Puede generar contenidos de mayor contexto (longitud).
- Puede escalar muchas más tareas.

Resultados en contra:

- Falta de información sobre el dataset con el que está entrenado para entender los parcialidad que puede tener.
- Sigue "alucinando".
- Tiende a escribir contenidos plausibles pero incorrectos con mucha seguridad.
- Solo puede ser usado vía endpoint de OpenAI por lo que uno es "esclavo" del producto.
- Modelo extremadamente costoso (OpenAI achica los costos para ganar usuarios).