Transformers y LLMs

La Arquitectura que Revolucionó la Inteligencia Artificial

Una exploración profunda de los fundamentos técnicos detrás de los modelos de lenguaje modernos y su impacto en el futuro de la IA

El Acertijo del Lenguaje

"El gato maulla y el perro..."

¿Qué hace el perro?

- ¿Ladra? ✓
- ¿Se asusta?
- ¿No maulla?

La mayoría responde: **"ladra"**

¿Cómo lo hicimos?

A través de algo que se llama **ATENCIÓN**. Le ponemos atención a ciertas palabras y a otras no.

Esto se puede expresar matemáticamente. Esta capacidad de "atención" es el corazón de los Transformers y la IA moderna.

1. Tokenización: Rompiendo el Lenguaje

El Primer Paso

Tomamos **todo el lenguaje de la cultura humana** y lo rompemos en pedacitos: letras, sílabas y palabras.

Tamaño de Vocabularios

- **Traductores:** 40,000 - 50,000 tokens
- **GPT-4, Llama, etc.:** hasta 256,000 tokens

Ejemplo: "satisfacción"

```
satisfacción → [SAT] [IS] [F] [ACCIÓN]

Cada pedazo es un TOKEN
```

Así manejamos palabras nuevas o raras dividiéndolas en partes conocidas.

2. Espacio N-Dimensional: Embeddings

Correlación Entre Tokens

Evaluamos **qué tan cercana está cada palabra con otras** en todo el lenguaje.

Ejemplo con "GATO":

📊 **Eje ANIMAL:** gato = 0 (muy cerca)

🚗 **Eje AUTOMÓVIL:** gato = 100 (muy lejos)

❤️ **Eje AMOR:** gato = 10 (algo cerca)

Palabras Como Vectores

```
Rey - Hombre + Mujer ≈ Reina
Italia - Roma + Colombia ≈ Bogotá
```

¡Podemos hacer matemáticas con significados!

3. El Problema: Modelos Secuenciales

Antes de 2017: RNNs y LSTMs

Los modelos dominantes procesaban texto **palabra por palabra, secuencialmente**:

❌ Modelos Antiguos (RNN/LSTM)

- Procesamiento secuencial
- Lento (no paralelizable)
- "Olvidan" el inicio en textos largos
- Difícil de entrenar

✓ Lo que Necesitábamos

- Procesamiento paralelo
- Rápido y eficiente
- Contexto completo siempre
- Escalable

La solución: Eliminar la recurrencia por completo y usar solo ATENCIÓN

4. "Attention is All You Need" (2017)

El Paper que Cambió Todo

Autores: Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser, Polosukhin (Google Brain)
Citado: Más de 173,000 veces (uno de los papers más citados del siglo XXI)

La Propuesta Revolucionaria

Una arquitectura simple basada **únicamente en mecanismos de atención**, eliminando completamente la recurrencia y convoluciones.

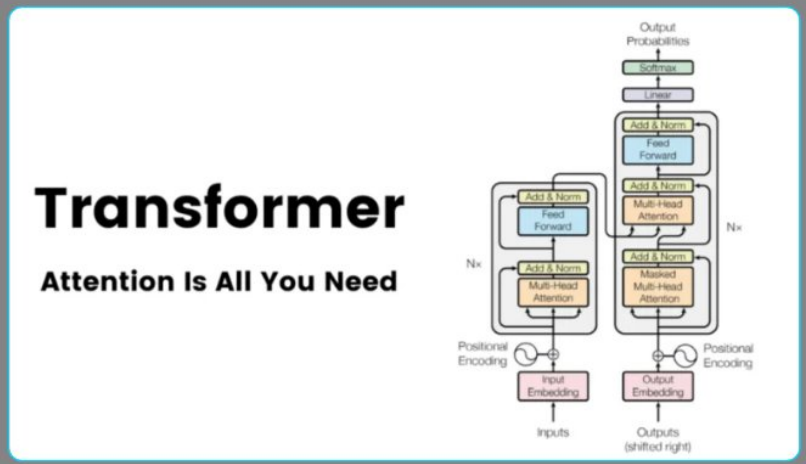
Resultados Iniciales

- **Traducción Inglés-Alemán:** 28.4 BLEU (superó todos los modelos anteriores)
- **Traducción Inglés-Francés:** 41.8 BLEU (nuevo récord)
- **Entrenamiento:** 3.5 días en 8 GPUs (fracción del costo de otros modelos)
- **Paralelización:** Significativamente más rápido

"Team Transformer" originalmente lo probaron en traducción, Wikipedia y análisis sintáctico, confirmando que era un modelo de lenguaje de propósito general.

5. Arquitectura del Transformer

Estructura Encoder-Decoder



ENCODER

Función: Procesa la entrada completa y genera representaciones contextuales

- Stack de 6 capas idénticas
- Cada capa tiene 2 sub-capas
- Bidireccional (ve todo el contexto)

Usado en: BERT, RoBERTa

DECODER

Función: Genera salida secuencialmente

- Stack de 6 capas idénticas
- Cada capa tiene 3 sub-capas
- Unidireccional (atención enmascarada)

Usado en: GPT, Claude, ChatGPT

6. Componentes Clave del Transformer

1. Multi-Head Self-Attention

El corazón del sistema. Permite que cada token "atienda" a todos los demás simultáneamente.

8 cabezas en paralelo - cada una aprende diferentes tipos de relaciones (sintaxis, semántica, etc.)

2. Position-wise Feed-Forward Networks

Aplica transformaciones no lineales a cada posición independientemente.

3. Positional Encoding

Inyecta información de posición usando funciones seno y coseno
Sin esto, el Transformer no sabría el orden de las palabras.

4. Residual Connections + Layer Normalization

Facilita el entrenamiento de redes profundas evitando el vanishing gradient.

7. Mecanismo de Atención (Detallado)

Scaled Dot-Product Attention

$$Attention(Q, K, V) = softmax(QK^T / \sqrt{d_k}) \times V$$

Los Tres Componentes

Q (Query): "¿Qué estoy buscando?" - La palabra actual
K (Key): "¿Qué información ofrezco?" - Todas las palabras
V (Value): "Información real" - Los valores a extraer

Ejemplo: "El gato maulla y el perro..."

1. **Query:** "perro" (última palabra)
2. **Keys cercanas:** "maulla" y "gato" (en el espacio vectorial)
3. **Cálculo:** QK^T genera scores de similitud
4. **Softmax:** Convierte scores en probabilidades
5. **Resultado:** Vector que apunta a "ladra" con 87% de probabilidad

10. De Transformers a LLMs Modernos

El Camino

1. **2017:** Transformer para traducción
2. **2018:** BERT (encoder) y GPT (decoder)
3. **2019-2020:** Escalamiento masivo (GPT-2, GPT-3)
4. **2022:** RLHF + ChatGPT (conversacional)
5. **2023-2024:** GPT-4, Claude, Gemini, LLaMA

RLHF: El Toque Final

Reinforcement Learning with Human Feedback

OpenAI contrató 6,000 personas para hablar con GPT y:

- Regañarlo cuando no se comportaba como chat
- Recompensarlo cuando sí lo hacía

Esto enseñó al modelo: cuándo parar, cómo estructurar respuestas, personalidad conversacional.

Impacto

Los Transformers han democratizado la IA y se usan en:

- Procesamiento de lenguaje (ChatGPT, Claude)
- Visión por computadora (Vision Transformers)

Resumen: Transformers + LLMs

La Revolución Transformer

1. **Tokenización:** Dividir el lenguaje en tokens
2. **Embeddings:** Vectores en espacio n-dimensional
3. **Attention Mechanism:** Enfocarse en lo relevante
4. **Multi-Head Attention:** Múltiples perspectivas simultáneas
5. **Transformer Architecture:** Encoder-Decoder con atención
6. **Entrenamiento Masivo:** Miles de millones de parámetros
7. **RLHF:** Comportamiento conversacional

"El gato maulla y el perro LADRA"

La Atención es Todo lo que Necesitas

"Attention is All You Need"

¡Gracias por tu atención!