Aprendizaje automático 2

Redes neuronales en PLN

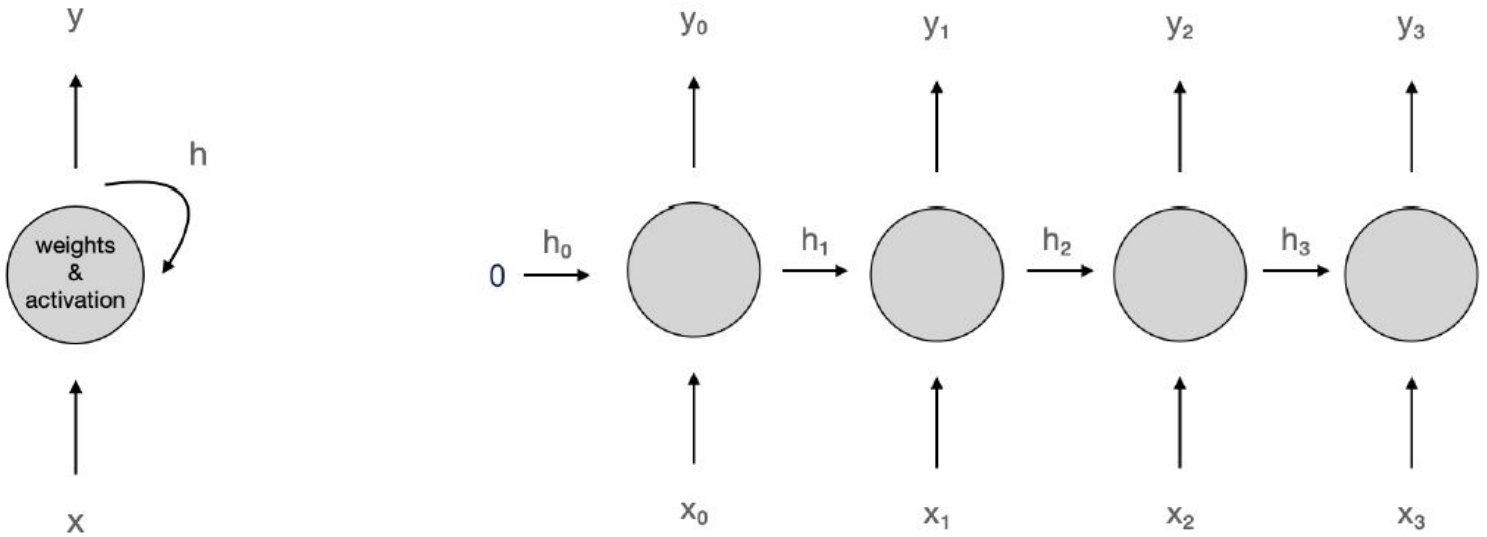
ISTEA

Redes para tratamiento secuencial

Redes recurrentes

- **RNN**: Recurrent Neural Networks
  - Entrada: x
  - Salida: y
  - Estado oculto: h

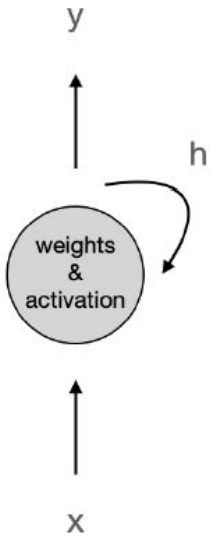
  Aplicaciones en tareas secuencia-a-secuencia, ej: RAH en tiempo real

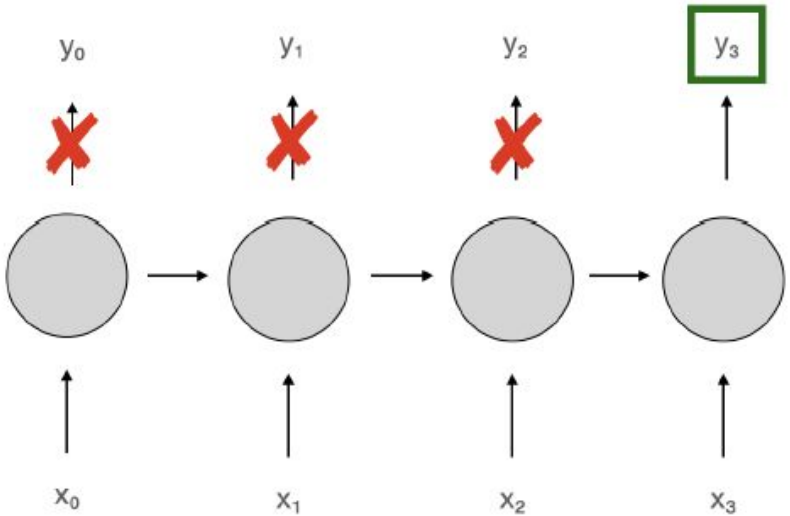


Redes recurrentes

- **RNN**: Recurrent Neural Networks
  - Entrada: x
  - Salida: y
  - Estado oculto: h

  Aplicaciones en tareas secuencia-a-vector, ej: clasificación de tópicos

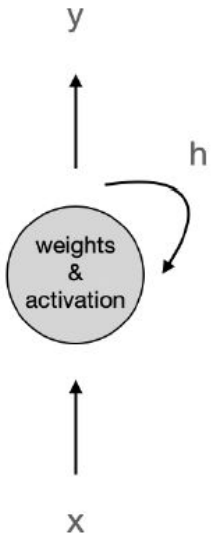


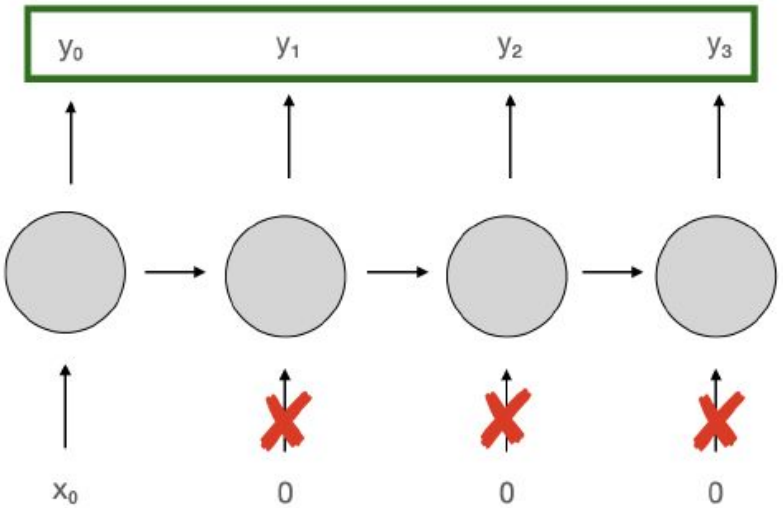


Redes recurrentes

- **RNN**: Recurrent Neural Networks
  - Entrada: x
  - Salida: y
  - Estado oculto: h

  Aplicaciones en tareas vector-a-secuencia, ej: generación de texto

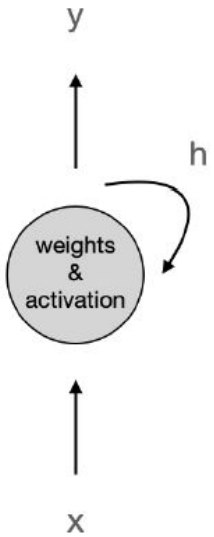


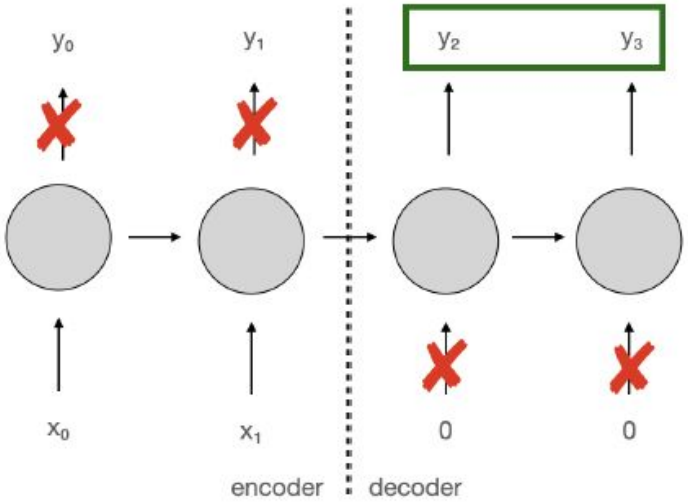


Redes recurrentes

- **RNN**: Recurrent Neural Networks
  - Entrada: x
  - Salida: y
  - Estado oculto: h

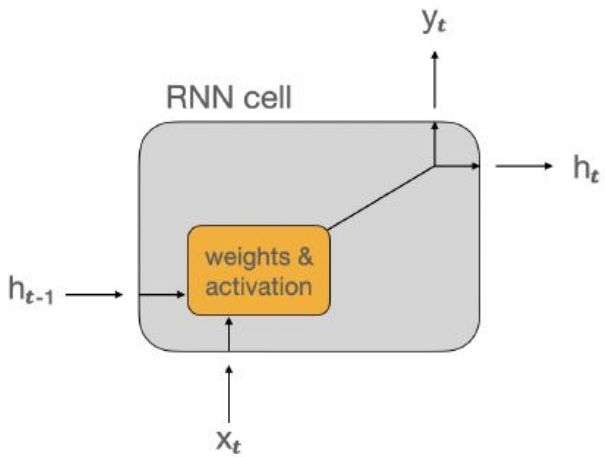
  Aplicaciones en tareas encoder-decoder, ej: traducción automática





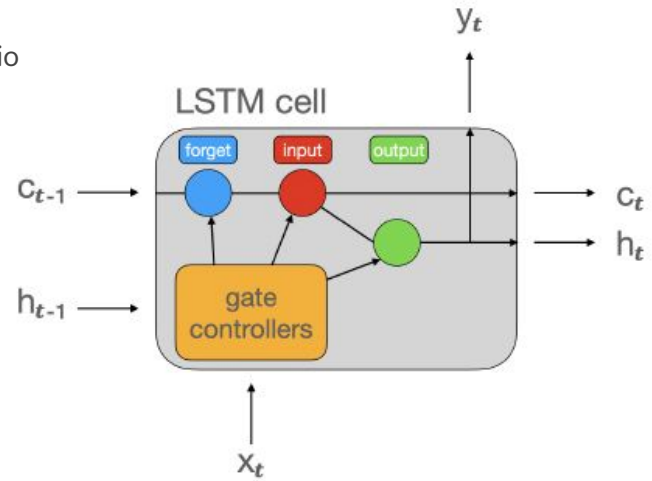
Redes recurrentes

- **RNN**: Recurrent Neural Networks
  - Representación alternativa
    - 2 entradas: X actual y h previo
    - 2 salidas: y actual y h para el próximo estado
  - Problema: memoria de corto plazo



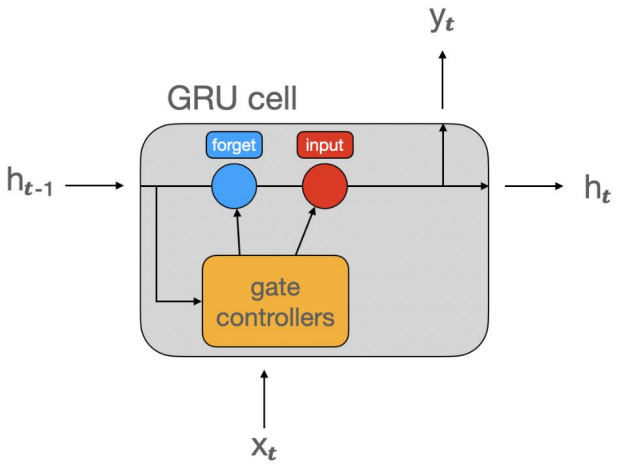
Redes recurrentes

- **LSTM**: Long short-term memory
  - Entradas y salidas:
    - 3 entradas: X actual, h previo y c (estado de largo plazo) previo
    - 2 salidas: y actual y h para el próximo estado
  - Compuertas internas:
    - “Forget”: qué remover de la memoria de largo plazo
    - “Input”: qué guardar para próximos estados
    - “Output”: qué devolver para el próximo estado



Redes recurrentes

- **GRU**: Gated Recurrent Unit
  - Versión simplificada de la LSTM.
  - Entradas y salidas:
    - 2 entradas: X actual, h previo
    - 2 salidas: y actual y h para el próximo estado
  - Compuertas internas:
    - “Forget”: qué remover de la memoria de largo plazo
    - “Input”: qué guardar para próximos estados



Modelos de lenguaje de gran escala (LLM: large language models)

¿Qué son los Transformers?

- Redes neuronales diseñadas para p
como texto o audio, de manera eficiente.
- Creadas por Google Brain en 2017.

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

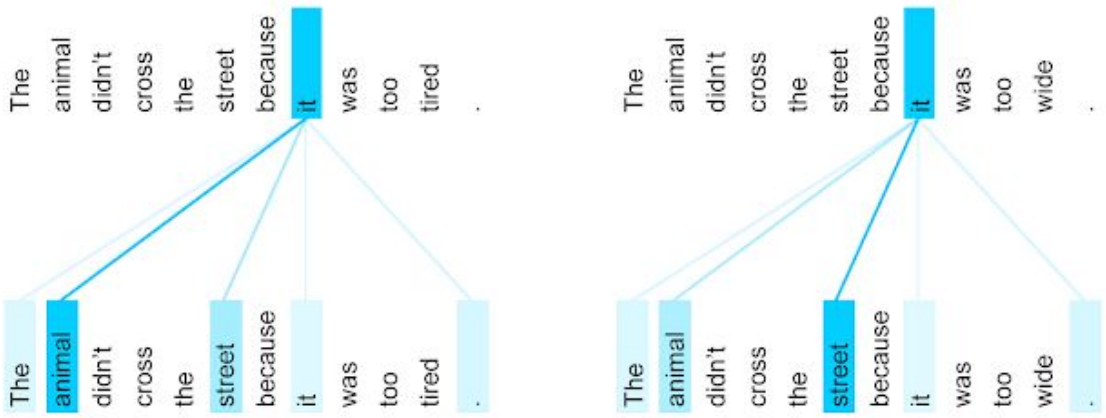
Fuente: [link]

¿Qué son los Transformers?

Desambiguación por la relación entre **todas** las palabras:

The animal didn't cross the street because it was too tired.
L'animal n'a pas traversé la rue parce qu'il était trop fatigué.

The animal didn't cross the street because it was too wide.
L'animal n'a pas traversé la rue parce qu'elle était trop large.



¿Qué son los Transformers?

Componentes Clave:

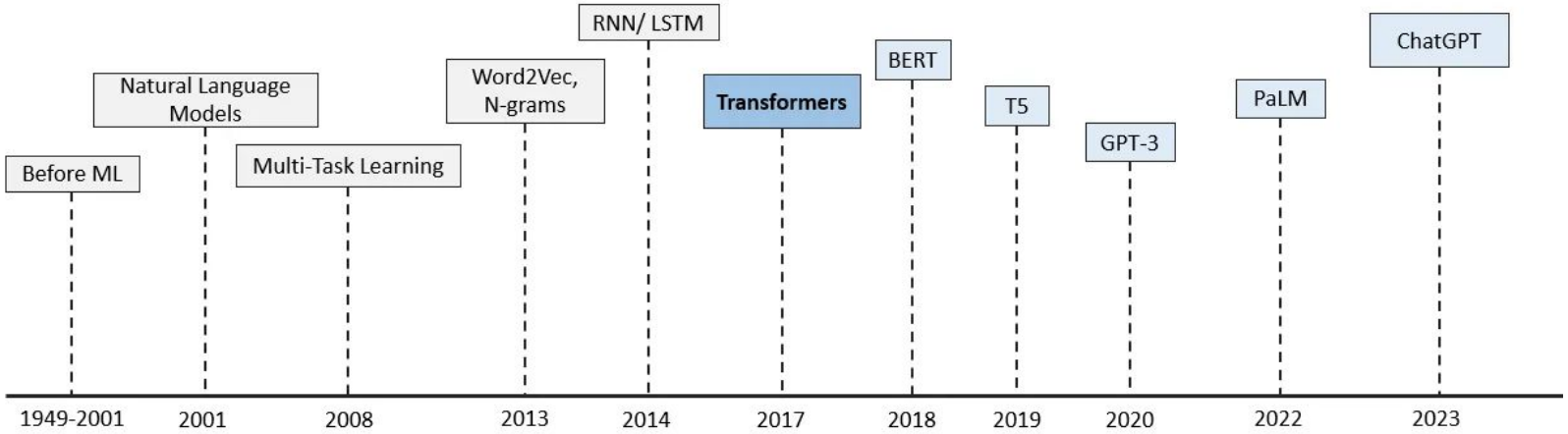
1. **Codificación Posicional**
   - Añade información sobre el orden de las palabras en una secuencia, ya que los Transformers no tienen un sentido de secuencialidad como las RNN.
     Ej: “El **libro** que leí ayer es mucho mejor que el **libro** que leí la semana pasada”

2. **Mecanismo de Atención**
   - Permite al modelo enfocarse en diferentes partes de la secuencia simultáneamente, calculando la relevancia de cada palabra con respecto a otras en la misma secuencia.
     Ej: “Aunque estaba lloviendo y la carretera estaba resbaladiza, el **conductor** del autobús, que tenía muchos años de experiencia y siempre mantenía la calma en situaciones difíciles, advirtió que un niño pequeño, que corría tras su pelota sin prestar atención al tráfico, se largó a cruzar la calle, pero afortunadamente **logró detener** el vehículo justo a tiempo antes de colisionarlo.”

3. **Procesamiento paralelo**
   - Al contrario de modelos previos, permite el entrenamiento e inferencia utilizando *toda* la potencia de las GPUs.

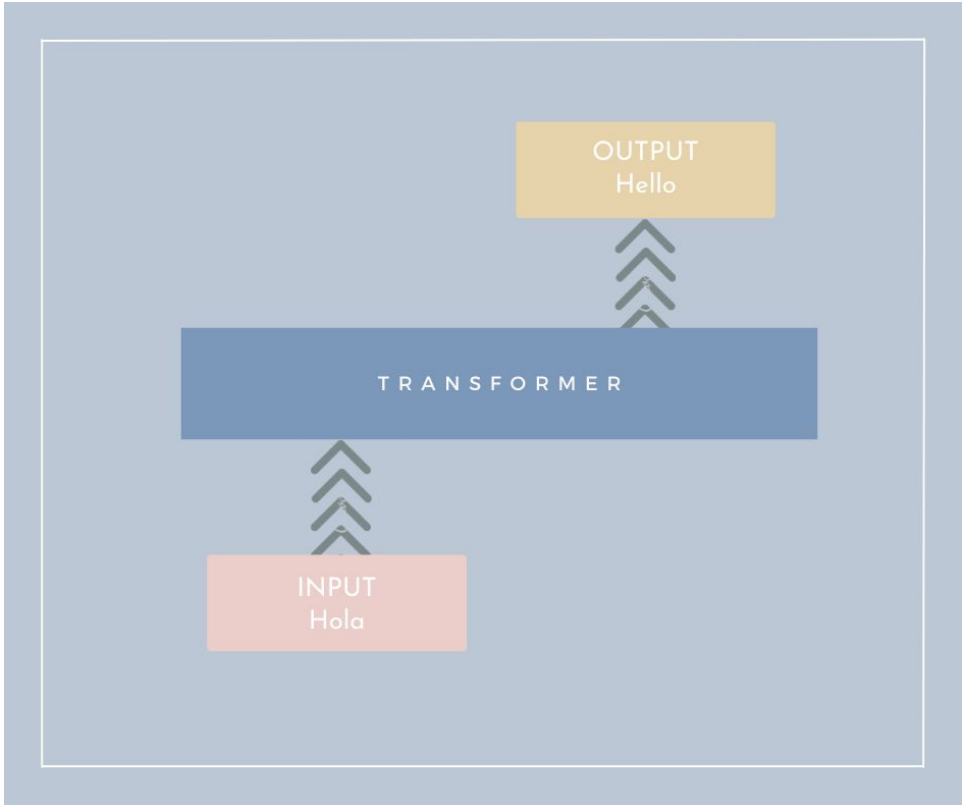
Relevancia y Evolución

- Línea de tiempo



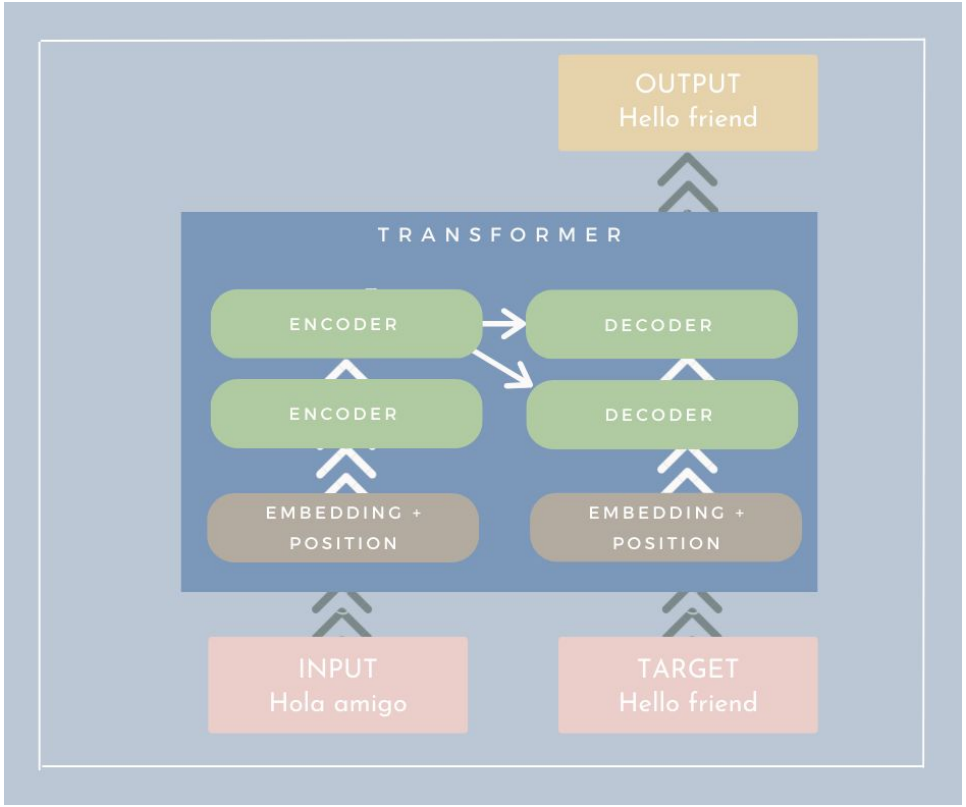
Fuente: [link]

Componentes

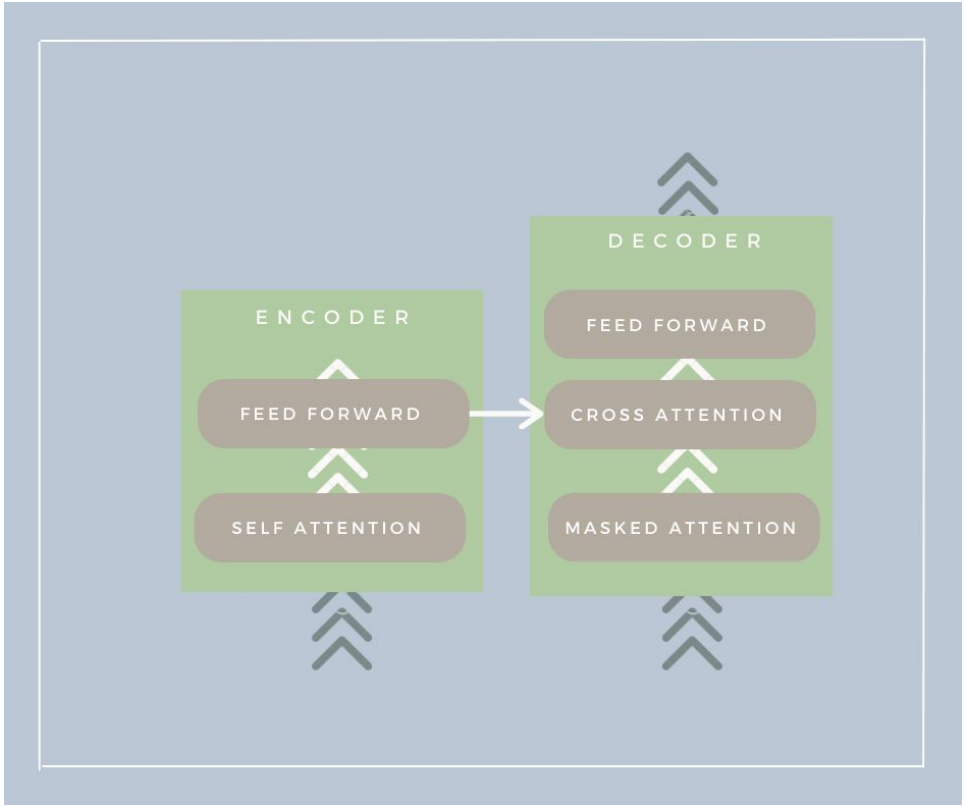


https://www.aprendemachinelearning.com/como-funcionan-los-transformers-espanol-nlp-gpt-bert/

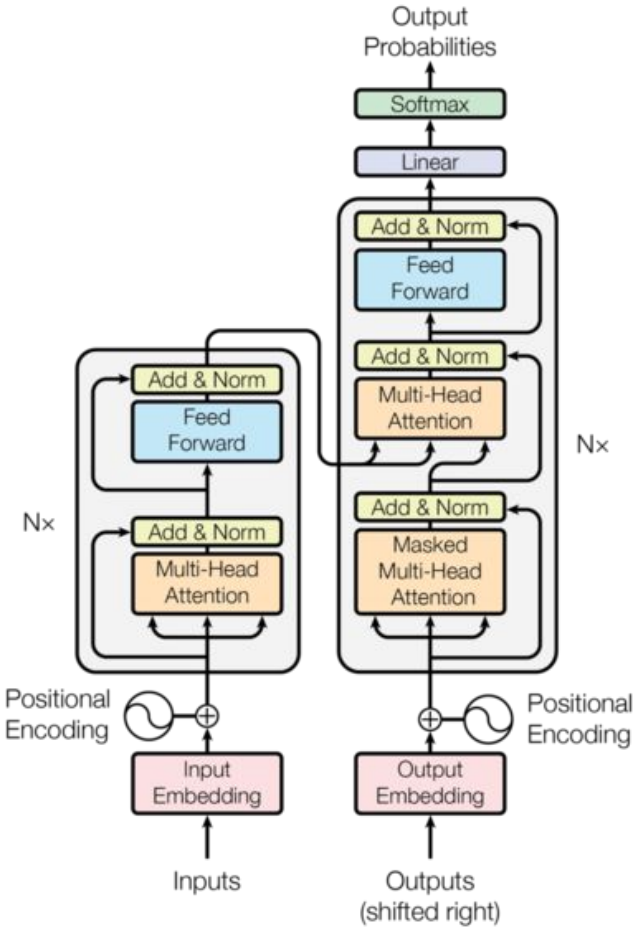
Componentes



Componentes



Componentes



Componentes

1. Embedding Layer (Capa de Embedding)
    - Convierte palabras en vectores numéricos de dimensiones fijas, capturando relaciones semánticas entre palabras.
2. Positional Encoding (Codificación Posicional)
    - Añade información sobre el orden de las palabras en una secuencia, ya que los Transformers no tienen un sentido de secuencialidad como las RNN.
3. Multi-Head Self-Attention (Atención de Múltiples Cabezas)
    - Permite al modelo enfocarse en diferentes partes de la secuencia simultáneamente, calculando la relevancia de cada palabra con respecto a otras en la misma secuencia.

Componentes

- Feed-Forward Neural Network (Red Neuronal de Alimentación Directa)
  Aplica transformaciones no lineales a las representaciones generadas por la capa de atención para capturar patrones más complejos.
- Add & Norm (Suma y Normalización)
  Suma la salida de la capa de atención con su entrada original (residual connection) y luego normaliza el resultado para estabilizar el entrenamiento.
- Encoder Layer (Capa del Codificador)
  Conjunto de capas que procesan la entrada y generan una representación codificada de la misma, repetido varias veces (generalmente 6 o 12).
- Output Layer (Capa de Salida)
  Proporciona las predicciones finales, como etiquetas de clasificación o palabras generadas, dependiendo de la tarea.

Modelos Principales

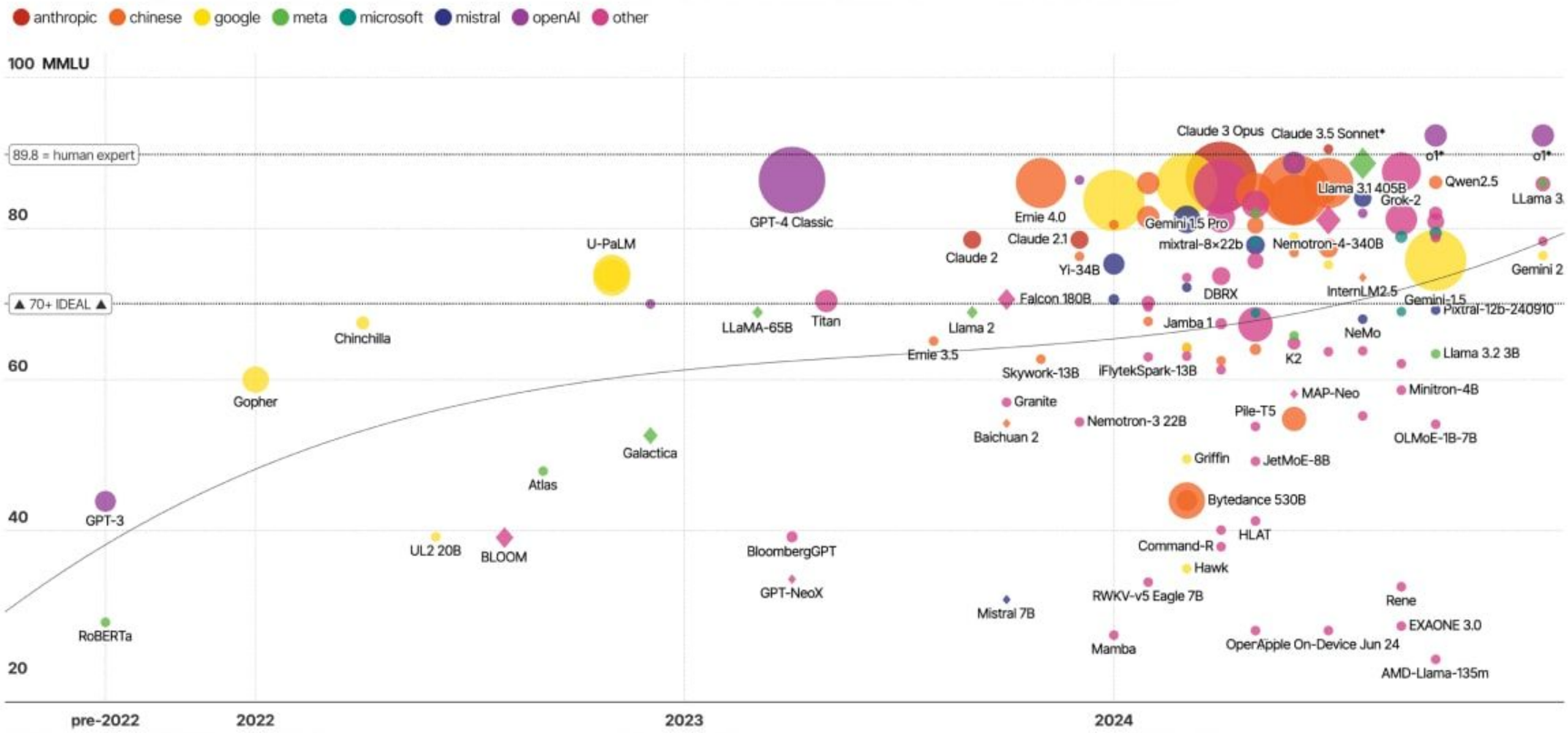
- Modelos solo Encoder: clasificación de texto, detección de sentimiento.
  - BERT: Predicción de tokens y relaciones bidireccionales.
  - RoBERTa: BERT optimizado para preentrenamiento.
  - DistilBERT: BERT reducido para uso en dispositivos con menos recursos.

- Modelos solo Decoder: generación de texto
  - GPT: Generación de texto autoregresiva.
  - GPT-2: Expansión de GPT con más capacidad.
  - GPT-3: Versión a gran escala de GPT para tareas generales de lenguaje.
  - GPT-4: Incorpora multimodalidad

  Modelos comerciales más famosos: chatGPT (OpenAI), LlaMA (Meta AI), Gemini (Google), Claude (Anthropic)

- Modelos completos (Encoder-Decoder): traducción automática, resúmenes y corrección de textos
  - BART (Meta AI): Combina codificación bidireccional y generación autoregresiva.
  - T5 (Google): Traducción, resumen y tareas secuencia a secuencia.
  - mT5 (Google): variante del T5 para mejor soporte multilingüe.

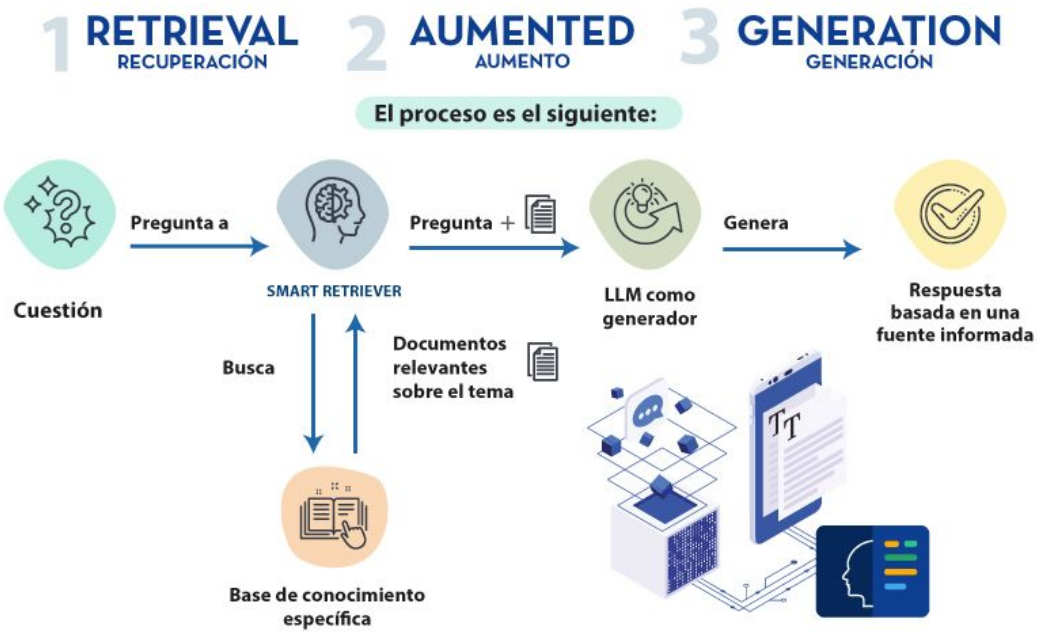
Explosión de modelos!



Avances recientes

Retrieval-Augmented Generation (RAG)

- Técnica que combina modelos de generación de texto (como transformers) con sistemas de recuperación de información.
- Aplicaciones: tareas de respuestas a preguntas (QA), asistentes virtuales, y generación de contenido basado en datos actualizados o específicos de un área de conocimiento.



Fuente: Adaptación de https://www.ml6.eu/blogpost/leveraging-llms-on-your-domain-specific-knowledge-base

Más innovaciones!

- Modelos de Multimodalidad: Integración de texto, imágenes, y audio en modelos como CLIP y DALL-E.
- InstructGPT y Modelos Ajustados con Instrucciones: Mejoras en la personalización y control de modelos generativos basados en instrucciones específicas.
- Avances en Fine-Tuning Eficiente: Nuevas técnicas como LoRA (Low-Rank Adaptation) que permiten ajustar grandes modelos con menos recursos.
- Modelos de Bajo Recurso: Enfoques para mejorar la eficacia de modelos en lenguajes con datos limitados.

- Modelos Generativos de Código por Vibe: sistemas como GitHub Copilot Studio, Claude Code, CodeGPT, que pueden generar código basado en descripciones de "vibra".
- LLMs Multimodales para Diseño de UI/UX: Modelos que traducen descripciones de "vibe" o capturas de pantalla en código funcional de frontend, como V-coder o Devin.

Medidas de desempeño

Métricas en LLM

Generación de Texto:

- BLEU (Bilingual Evaluation Understudy): Mide la calidad de las traducciones generadas comparándolas con referencias humanas.
- ROUGE (Recall-Oriented Understudy for Gisting Evaluation): Evalúa la calidad de resúmenes comparando con resúmenes de referencia, basado en la superposición de n-gramas.
- METEOR (Metric for Evaluation of Translation with Explicit ORdering): Evalúa la calidad de la traducción considerando la concordancia de palabras, sinónimos y el orden de las palabras.

Traducción Automática:

- BLEU: Para evaluar la calidad de las traducciones generadas.
- TER (Translation Edit Rate): Mide el número de ediciones necesarias para convertir una traducción generada en una traducción de referencia.

Bibliografía

- Goyal, P., Pandey, S., & Jain, K. (2018). Deep learning for natural language processing. *New York: Apress*. [link]
- Liu, Y., & Zhang, M. (2018). Neural network methods for natural language processing. [link]
- Tunstall, L., Von Werra, L., & Wolf, T. (2022). *Natural language processing with transformers*. " O'Reilly Media, Inc.". [link]
- Repositorio HuggingFace [link] - Playground [link]
- Librería Langchain [link]

- “Transformers (how LLMs work) explained visually”, https://youtu.be/wjZofJX0v4M