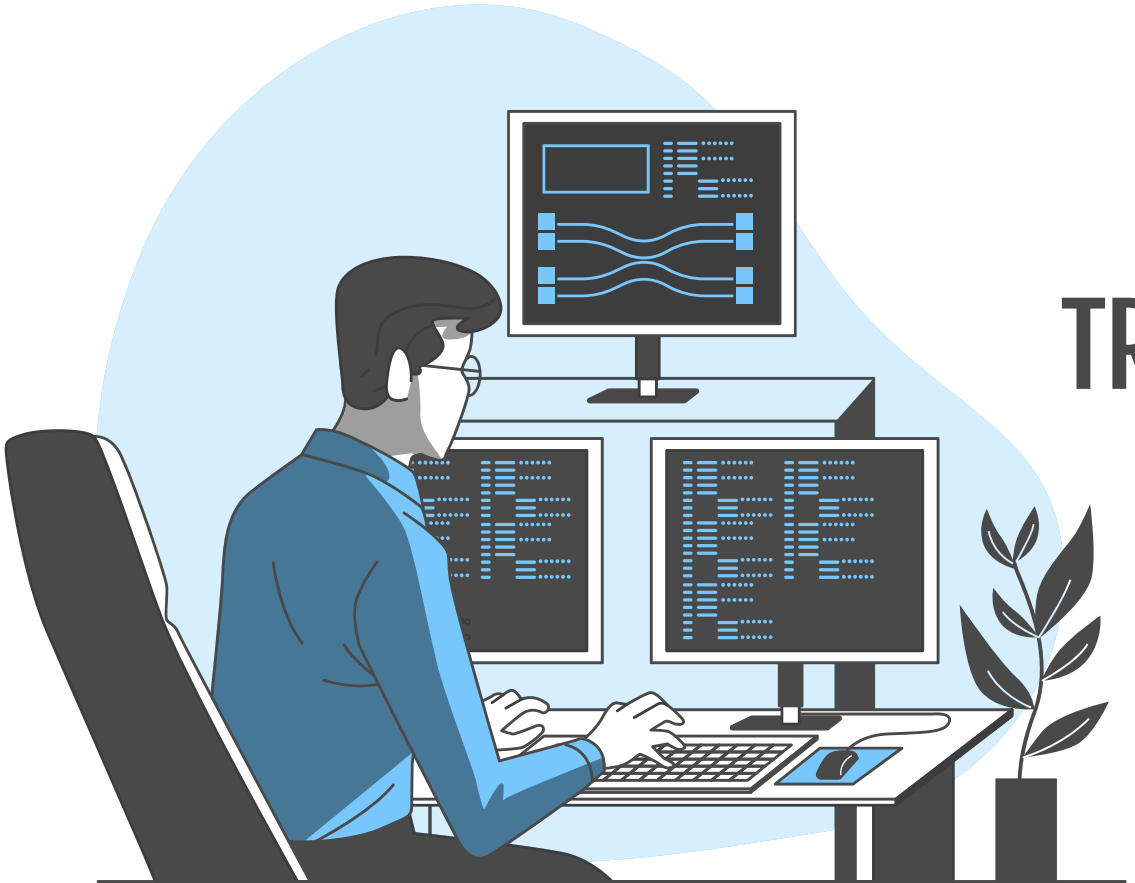
IPAR
ESTUDIOS



NLP Y TRANSFORMERS

¿Qué es el Procesamiento de Lenguaje Natural (NLP)?

Definición

El **Procesamiento de Lenguaje Natural (NLP)** es un campo interdisciplinario que combina:

- **Ciencias de la Computación**
- **Inteligencia Artificial**
- **Lingüística**

Objetivo: Estudiar las interacciones entre computadores y el lenguaje humano

Desafíos del NLP

- **Ambigüedad**: Una palabra puede tener múltiples significados
- **Contexto**: El significado depende del contexto
- **Variabilidad**: Diferentes formas de expresar la misma idea
- **Cultura y región**: Expresiones idiomáticas y regionalismos

Evolución Histórica: El Camino hacia los Transformers

Años 1990-2010: Machine Learning Clásico

- Modelos de Markov
- Support Vector Machines
- Naive Bayes

Años 2010-2017: Redes Neuronales Recurrentes

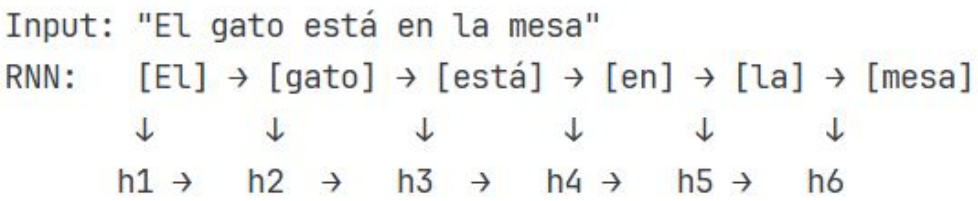
- **RNNs**: Procesamiento secuencial
- **LSTMs**: Memoria a largo plazo limitada
- **Seq2Seq**: Modelos encoder-decoder

2017-Presente: Era de los Transformers

- **"Attention is All You Need"** (Vaswani et al., 2017)
- Procesamiento paralelo
- Mecanismos de atención avanzados

Limitaciones de las RNNs

⏳ Procesamiento Secuencial



Problemas:

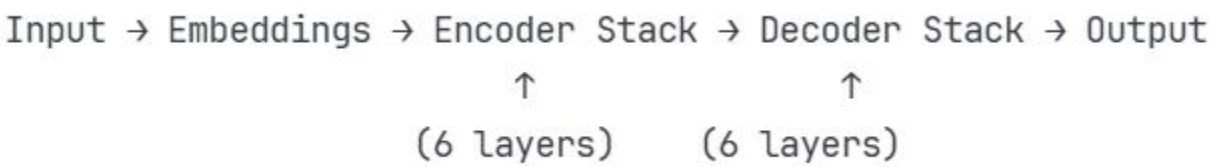
- **Lento**: No se puede paralelizar
- **Memoria limitada**: Gradientes que desaparecen
- **Dependencias largas**: Dificultad para recordar información lejana

🚫 Limitaciones técnicas

- **Vanishing gradients**: Pérdida de información en secuencias largas
- **Bottleneck**: Toda la información debe pasar por un estado oculto
- **Recursos computacionales**: No aprovecha GPUs eficientemente

Arquitectura Transformer: "Attention is All You Need"

🏗️ Visión General



Componentes Principales

1. Embeddings + Positional Encoding

- Convierte palabras en vectores numéricos (dimensión 512)
- Añade información posicional usando funciones seno/coseno

2. Encoder Stack (6 capas)

- Multi-Head Self-Attention
- Feed-Forward Networks
- Residual Connections + Layer Normalization

3. Decoder Stack (6 capas)

- Masked Multi-Head Self-Attention
- Multi-Head Cross-Attention
- Feed-Forward Networks
- Residual Connections + Layer Normalization

Encoder y Decoder en Transformers

Encoder

- **Procesa la entrada**: Convierte el texto de entrada en representaciones numéricas
- **Comprende el contexto**: Analiza relaciones entre todas las palabras simultáneamente
- **Self-attention**: Cada palabra "atiende" a todas las demás para entender su significado

Decoder

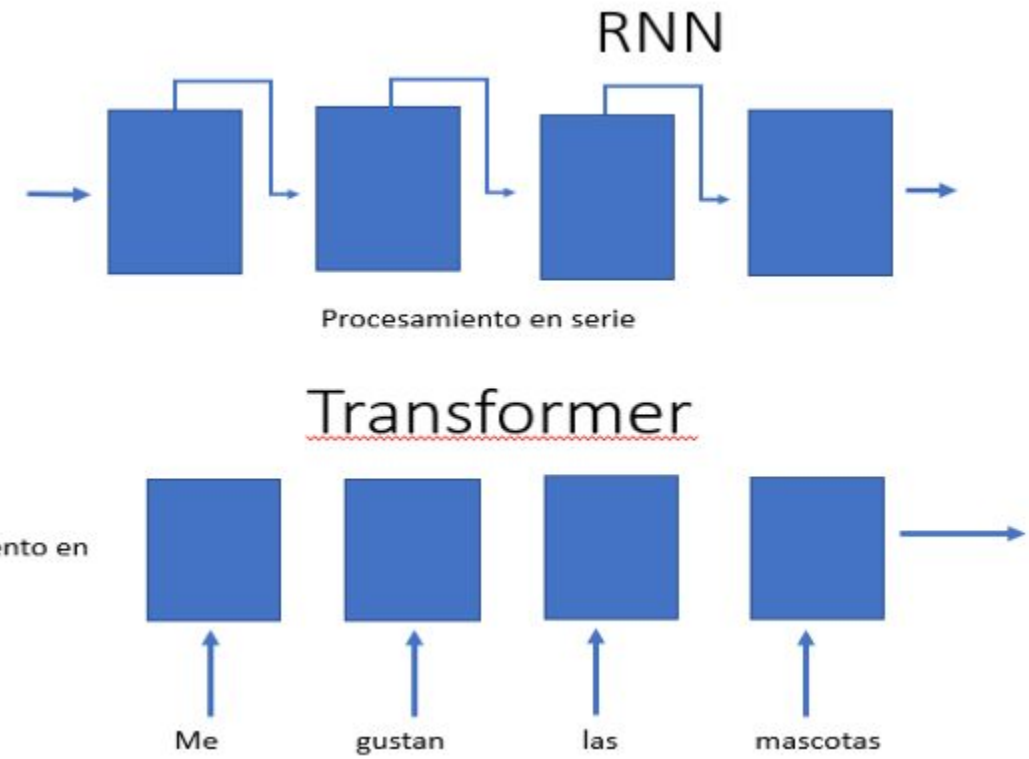
- **Genera la salida**: Produce el texto de respuesta palabra por palabra
- **Usa información del encoder**: Se basa en las representaciones creadas por el encoder
- **Atención cruzada**: Conecta la entrada procesada con la salida que se está generando

Ejemplo práctico

Traducción: "Hello world" → "Hola mundo"

- **Encoder**: Entiende "Hello world"
- **Decoder**: Genera "Hola mundo" basándose en esa comprensión

Arquitectura Transformer: "Attention is All You Need"



Mecanismo de Atención: El Corazón del Transformer

🧠 ¿Qué es la Atención?

La **atención** permite al modelo enfocarse en diferentes partes de la entrada según su relevancia para la tarea actual.

Self-Attention: Paso a Paso

Paso 1: Crear Query, Key, Value

```
# Para cada palabra, creamos 3 vectores:
Q = Query   # "¿Qué estoy buscando?"
K = Key     # "¿Qué información tengo?"
V = Value   # "¿Cuál es la información real?"
```

Paso 2: Calcular Attention Scores

```
# Producto punto entre Query y Key
scores = Q × K^T / √(dim_k)
```

Paso 3: Aplicar Softmax

```
# Convierte scores en probabilidades
attention_weights = softmax(scores)
```

Paso 4: Ponderar Values

```
# Combina información según importancia
output = attention_weights × V
```

Mecanismo de Atención: El Corazón del Transformer

📊 Ejemplo Visual: "El gato come pescado"

```
Palabra: "come"
Query de "come" → busca: sujeto y objeto

Attention weights:
- "El": 0.1    (artículo, poca importancia)
- "gato": 0.7  (sujeto, muy importante)
- "come": 0.1  (autorreferencia)
- "pescado": 0.1 (objeto, importante pero menos que sujeto)
```

Multi-Head Attention: Múltiples Perspectivas

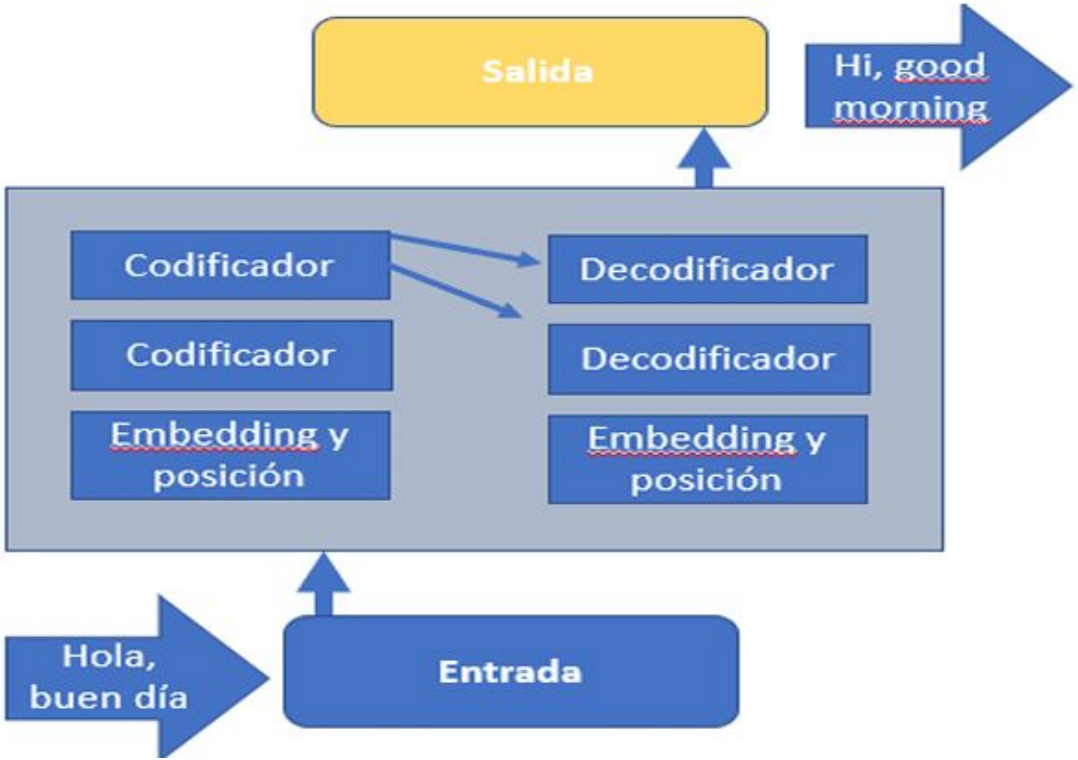
🔍 ¿Qué es?

"Multi-Head Attention" (atención multi-cabezal) es un mecanismo clave en los modelos Transformer, que permite capturar dependencias y relaciones entre palabras o tokens desde múltiples perspectivas al mismo tiempo.

Diferentes "cabezas" pueden enfocarse en diferentes tipos de relaciones:

- **Head 1**: Relaciones sintácticas (sujeto-verbo)
- **Head 2**: Relaciones semánticas (causa-efecto)
- **Head 3**: Dependencias a larga distancia

Esquema de un Transformer



Ventajas de los Transformers

⚡ Procesamiento Paralelo

- **RNN**: Secuencial → Lento
- **Transformer**: Paralelo → Rápido

🧠 Memoria a Largo Plazo

- Puede relacionar palabras muy distantes en el texto
- No sufre de vanishing gradients

🎯 Atención Selectiva

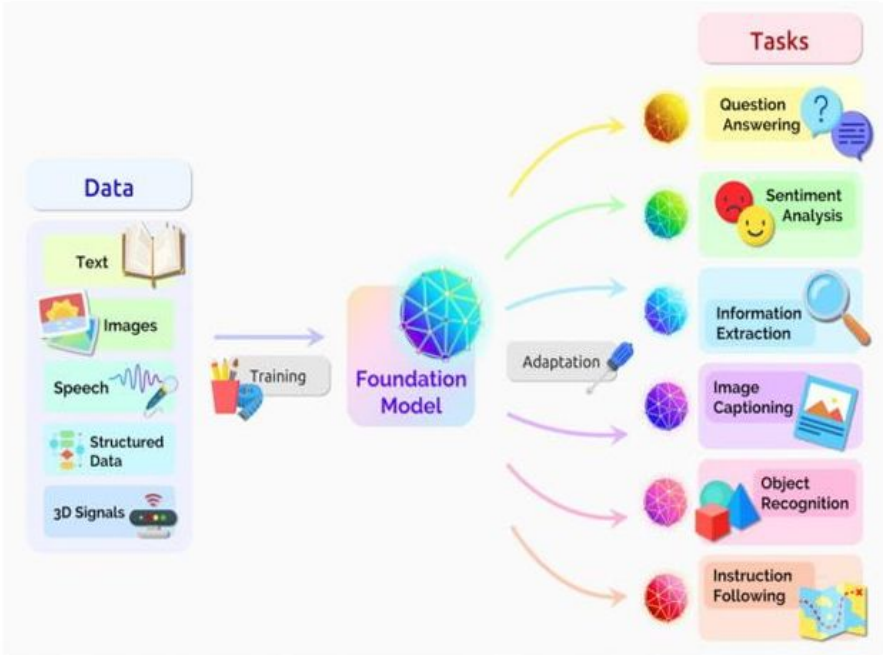
- Se enfoca en información relevante
- Ignora ruido irrelevante

🔧 Versatilidad

- Traducción automática
- Generación de texto
- Análisis de sentimientos
- Respuesta a preguntas
- Y mucho más...

APLICACIONES

1. Procesamiento de Lenguaje Natural (NLP)
2. Respuesta a Preguntas (Question Answering)
3. Sistemas de Chatbots y Asistentes Virtuales
4. Traducción Automática
5. Resumen Automático de Textos
6. Corrección Automática y Sugerencias
7. Detección de Spam y Fake News
8. Extracción Automática de Información
9. SEO y Marketing Digital
10. Análisis y Minería de Opiniones
11. Generación y Evaluación de Textos
12. Análisis de Correos Electrónicos



Modelos Transformer Famosos

📊 Comparación de Modelos

Modelo	Organización	Parámetros	Especialidad
GPT-3	OpenAI	175B	Generación de texto
GPT-4	OpenAI	~100T	Multimodal
BERT	Google	345M	Comprensión
BARD	Google	137B	Conversación
Megatron-Turing	Microsoft/NVIDIA	530B	Investigación

🎯 Arquitecturas Especializadas

- **BERT**: Bidirectional Encoder (solo encoder)
- **GPT**: Generative Pre-trained (solo decoder)
- **T5**: Text-to-Text Transfer (encoder-decoder completo)
- **Vision Transformer**: Para procesamiento de imágenes

¿Qué es BERT?

BERT (Bidirectional Encoder Representations from Transformers)

- Modelo de lenguaje desarrollado por Google (2018)
- Comprende el contexto de las palabras en ambas direcciones
- Pre-entrenado en grandes cantidades de texto
- Revolucionó el procesamiento de lenguaje natural (NLP)
- Aplicaciones: búsquedas, chatbots, análisis de sentimientos

Modelos Bert

🚀 Modelos BERT clásicos y variaciones

Modelo	Tarea típica	Descripción breve
`bert-base-uncased`	General NLP	BERT original (lowercase inglés)
`bert-large-uncased`	General NLP	BERT más grande (24 capas)
`bert-base-cased`	General NLP	Mantiene mayúsculas (cased)
`bert-base-multilingual-cased`	General NLP multilingüe	~104 idiomas
`bert-base-chinese`	Chino	Tokenización de caracteres chinos

Modelos Bert

🚀 1️⃣ Clasificación de texto (**Text Classification**)

👉 **¿Qué hace?**

Asigna una etiqueta (o varias) a un texto completo.

🔹 Ejemplos:

- **Análisis de sentimiento:** determinar si un texto es positivo, negativo o neutro.
- **Clasificación temática:** etiquetar noticias en categorías (deportes, política, economía).
- **Detección de spam:** ¿Es spam o no?

🚀 Modelos para **clasificación de texto en general**

Modelo	Particularidad
`cardiffnlp/twitter-roberta-base-sentiment`	Clasificación de sentimiento en tweets
`textattack/bert-base-uncased-imdb`	Sentimiento binario en IMDB

Modelos Bert

🚀 2️⃣ Clasificación de tokens (**Token Classification**)

👉 ¿Qué hace?

Asigna una etiqueta a cada palabra o token. Ej: detección de entidades nombradas (NER).

🔹 Ejemplos:

- **NER:** reconocer personas, organizaciones, lugares en un texto.
- Ejemplo:

```
"Elon Musk fundó SpaceX en 2002."
 PER       ORG
```

🚀 Modelos para **NER (Reconocimiento de Entidades)**

Modelo	Idioma / Dataset	Tarea
`dslim/bert-base-NER`	Inglés (CoNLL)	Etiquetado BIO de personas, orgs, locs, misc
`Davlan/bert-base-multilingual-cased-ner-hrl`	Multilingüe NER	~40 idiomas

Modelos Bert

🔥 2. Análisis de sentimiento (**Sentiment Analysis**)

- **¿Qué hace?**: Determina el tono o polaridad del texto (positivo, negativo, neutro o escalas más detalladas).
- Ejemplos:
  - Opiniones de productos: «Muy bueno» → Positivo
  - Reviews en Amazon, Yelp, Twitter.

🚀 Modelos fine-tuned para **análisis de sentimiento**

Modelo	Idioma / Dataset	Salida típica
`nlptown/bert-base-multilingual-uncased-sentiment`	Multi-idioma, reseñas	1-5 estrellas
`distilbert-base-uncased-finetuned-sst-2-english`	Inglés (SST-2)	POSITIVE / NEGATIVE
`finiteautomata/bertweet-base-sentiment-analysis`	Inglés, tweets	Sentiment Twitter

Modelos Bert

🚀 3️⃣ Respuesta a preguntas (**Question Answering / QA**)

👉 ¿Qué hace?

Le das un **contexto** + una **pregunta**, y el modelo extrae la respuesta directamente del texto.

- Ejemplo:

```
context = "Elon Musk fundó SpaceX en 2002."
question = "¿Cuándo fue fundada SpaceX?"
```

🚀 Modelos específicos para **preguntas y respuestas (QA)**

Modelo	Task
`deepset/bert-base-cased-squad2`	QA sobre SQuAD 2.0
`distilbert-base-cased-distilled-squad`	DistilBERT para QA