

Introducción al NLP y LLM's

Introducción a la IA

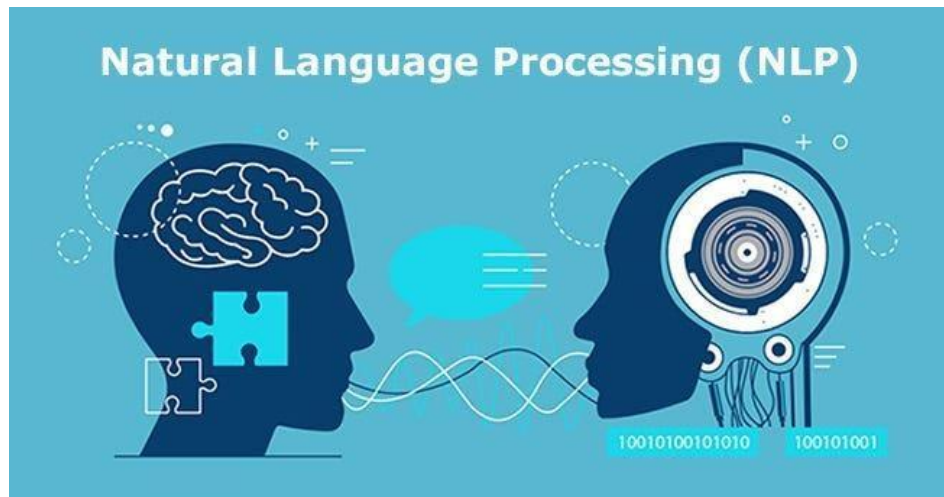
2025

Agenda

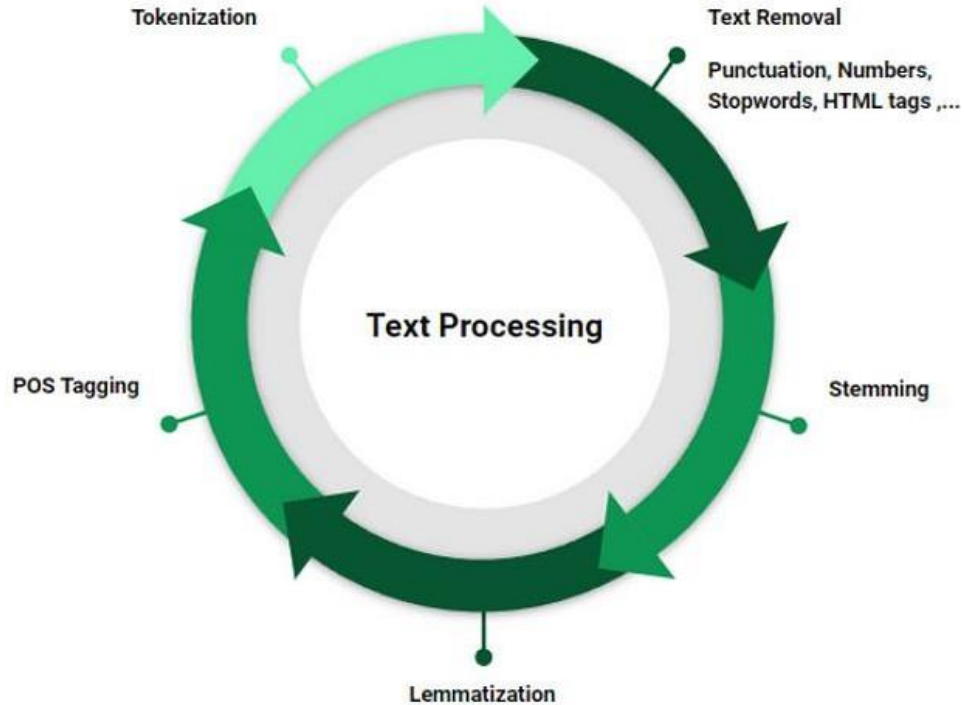
- Introducción a NLP
- Aplicaciones de NLP
- NLP con transformers
- Ejemplo práctico de Q&A usando Hugging Face.

Introducción a NLP

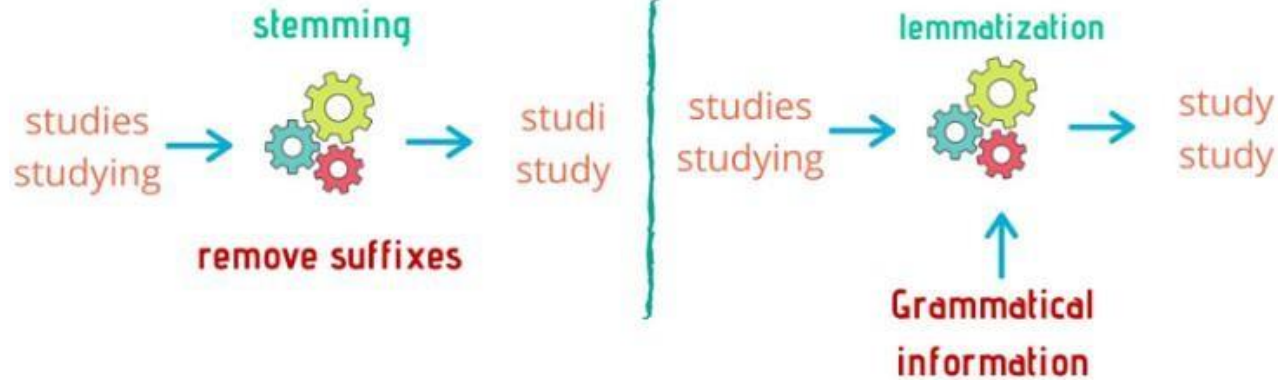
El Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés) es una rama de la inteligencia artificial que se enfoca en la interacción entre computadoras y el lenguaje humano. NLP tiene como objetivo ayudar a las computadoras a entender, interpretar y generar lenguaje de una manera que sea útil para tareas como la traducción automática, el análisis de sentimientos, la generación de texto, etc.



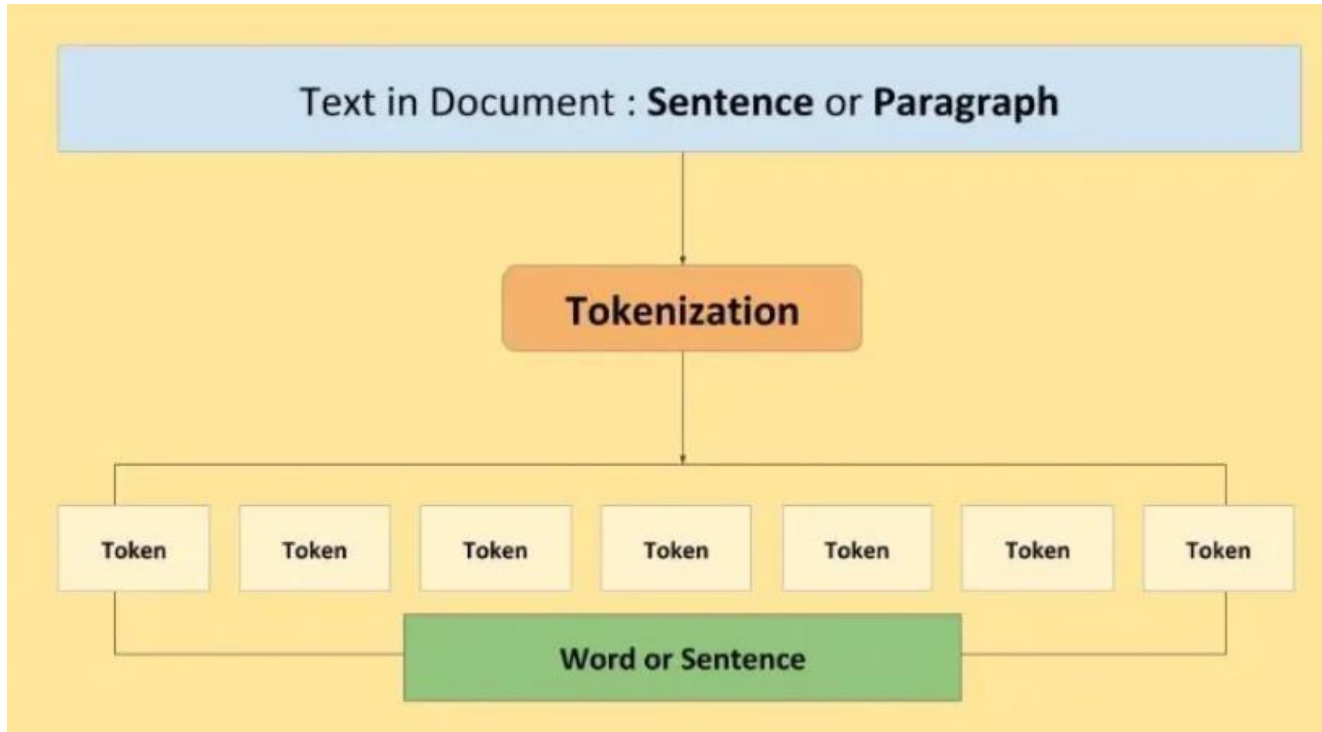
Introducción a NLP



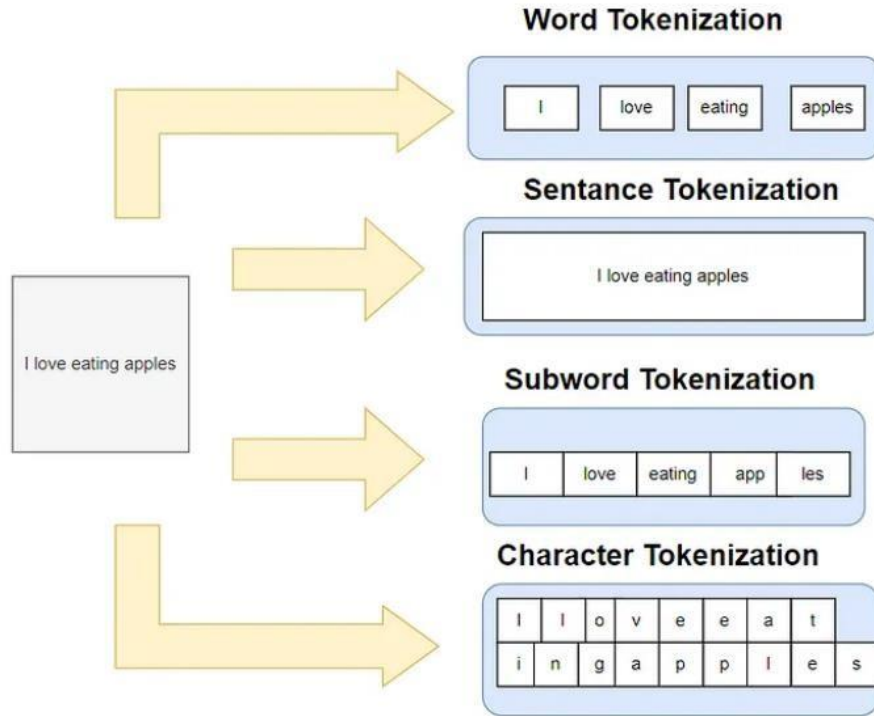
Stemming and Lemmatization



Tokenization

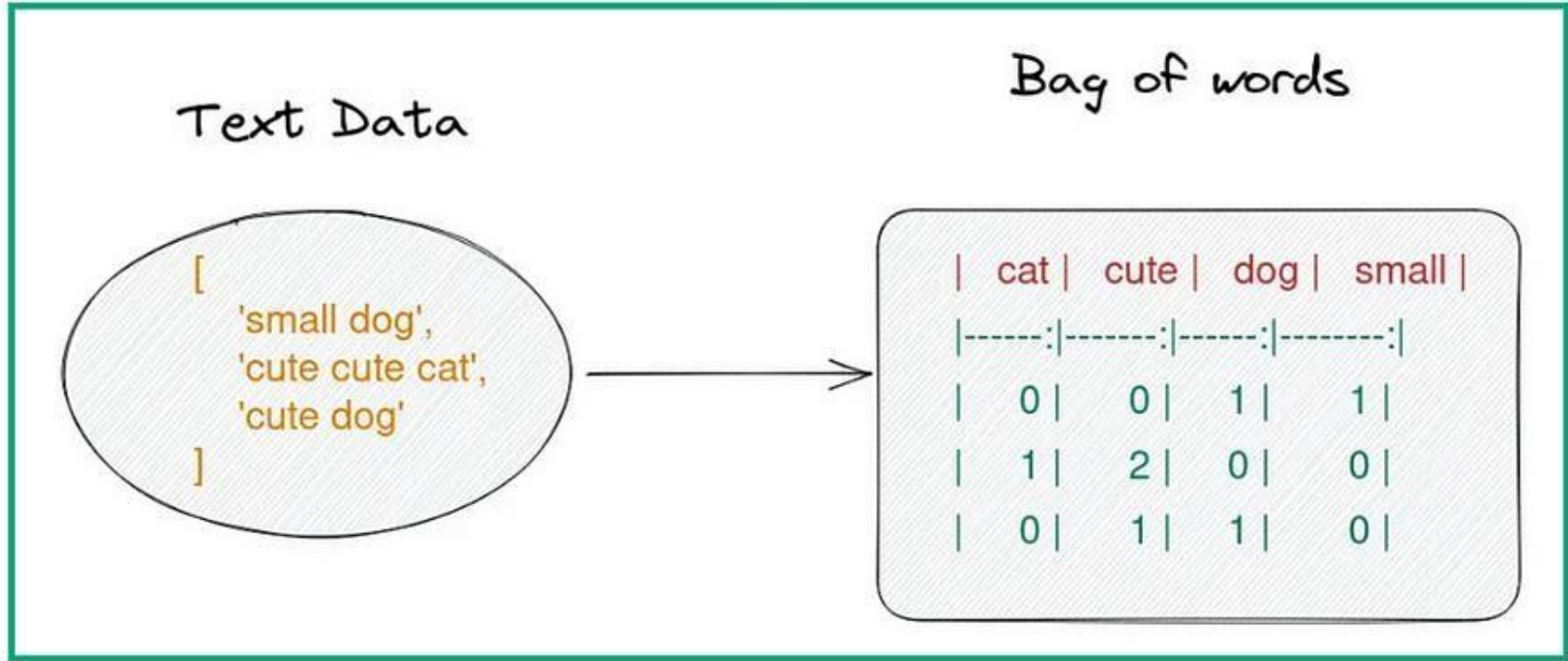


Tokenization



Tokenization techniques

Bag of words

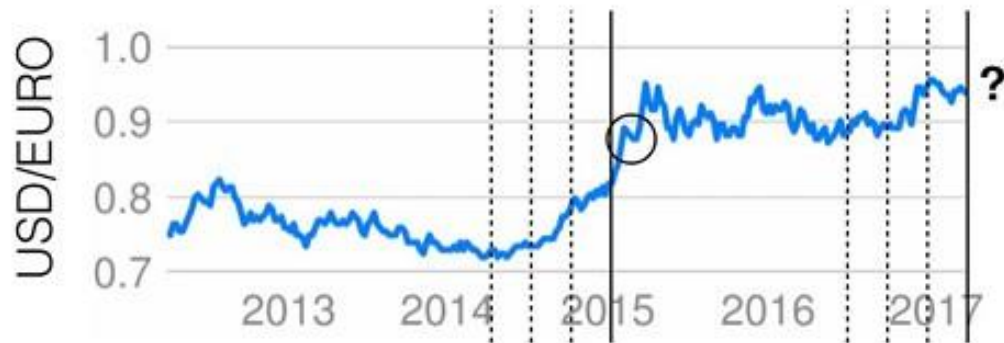


Bag of words

EJEMPLO BoW

DATOS SECUENCIALES

Cómo crear un problema de aprendizaje supervisado?



$$\begin{bmatrix} 0.82 \\ 0.80 \\ 0.73 \\ 0.72 \end{bmatrix} \quad 0.89$$

$\phi(t)$ $y(t)$

Los datos se pueden almacenar en vectores de características y “targets” utilizando ventanas deslizantes

DATOS SECUENCIALES

Modelado de lenguaje: qué viene después?

This course has been a tremendous|...

tremendous

$$\begin{bmatrix} 0 \\ \vdots \\ 1 \\ 0 \end{bmatrix}$$

?

a

$$\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$\phi(t)$

$y^{(t)}$

DATOS SECUENCIALES



Collections of elements where:

- Elements can be **repeated**
- **Order** matters
- Of **variable** (potentially infinite) length

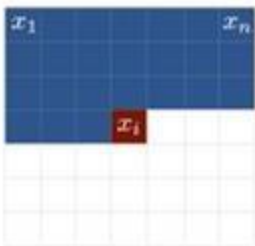
DATOS SECUENCIALES

- Los problemas de predicción de secuencias se pueden formular para entrenar una red neuronal feed-forward.
- Sin embargo, tenemos que ingeniar como mapear los datos históricos a un vector.
 - Cuántos pasos atrás debemos considerar?
 - Cómo mantener ítems importantes mencionados anteriormente?
- Alternativamente, nos gustaría aprender como codificar la “historia” de la secuencia en un vector.

DATOS SECUENCIALES

"Sequences really seem to be everywhere! We should learn how to model them. What is the best way to do that? Stay tuned!"

Words, letters



Images



1 Second

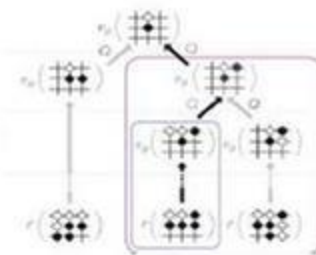
Speech



Videos

```
1 import numpy as np
2 import tensorflow as tf
3
4 def forward_backward_passes, H1:
5     H1 = tf.nn.conv2d(
6         H0, [tf.nn.conv2d],
7         [tf.nn.conv2d], [tf.nn.conv2d])
8
9     H1 = tf.nn.conv2d(
10        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
11
12     H1 = tf.nn.conv2d(
13        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
14
15     H1 = tf.nn.conv2d(
16        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
17
18     H1 = tf.nn.conv2d(
19        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
20
21     H1 = tf.nn.conv2d(
22        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
23
24     H1 = tf.nn.conv2d(
25        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
26
27     H1 = tf.nn.conv2d(
28        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
29
30     H1 = tf.nn.conv2d(
31        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
32
33     H1 = tf.nn.conv2d(
34        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
35
36     H1 = tf.nn.conv2d(
37        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
38
39     H1 = tf.nn.conv2d(
40        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
41
42     H1 = tf.nn.conv2d(
43        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
44
45     H1 = tf.nn.conv2d(
46        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
47
48     H1 = tf.nn.conv2d(
49        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
50
51     H1 = tf.nn.conv2d(
52        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
53
54     H1 = tf.nn.conv2d(
55        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
56
57     H1 = tf.nn.conv2d(
58        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
59
60     H1 = tf.nn.conv2d(
61        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
62
63     H1 = tf.nn.conv2d(
64        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
65
66     H1 = tf.nn.conv2d(
67        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
68
69     H1 = tf.nn.conv2d(
70        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
71
72     H1 = tf.nn.conv2d(
73        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
74
75     H1 = tf.nn.conv2d(
76        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
77
78     H1 = tf.nn.conv2d(
79        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
80
81     H1 = tf.nn.conv2d(
82        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
83
84     H1 = tf.nn.conv2d(
85        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
86
87     H1 = tf.nn.conv2d(
88        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
89
90     H1 = tf.nn.conv2d(
91        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
92
93     H1 = tf.nn.conv2d(
94        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
95
96     H1 = tf.nn.conv2d(
97        H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
98
99     H1 = tf.nn.conv2d(
100       H1, [tf.nn.conv2d], [tf.nn.conv2d], [tf.nn.conv2d])
```

Programs



Decision making

DATOS SECUENCIALES

	Supervised learning	Sequence modelling
Data	$\{x, y\}_i$	$\{x\}_i$
Model	$y \approx f_{\theta}(x)$	$p(x) \approx f_{\theta}(x)$
Loss	$\mathcal{L}(\theta) = \sum_{i=1}^N l(f_{\theta}(x_i), y_i)$	$\mathcal{L}(\theta) = \sum_{i=1}^N \log p(f_{\theta}(x_i))$
Optimisation	$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta)$	$\theta^* = \arg \max_{\theta} \mathcal{L}(\theta)$

DATOS SECUENCIALES

“Modeling word probabilities is really difficult”

DATOS SECUENCIALES

Simplest model:

Assume independence of words

$$p(\mathbf{x}) = \prod_{t=1}^T p(x_t)$$

$$p(\text{"modeling"}) \times p(\text{"word"}) \times p(\text{"probabilities"}) \times p(\text{"is"}) \times p(\text{"really"}) \times p(\text{"difficult"})$$

Word	$p(x_i)$
the	0.049
be	0.028
...	...
really	0.0005
...	...

However:

Most likely 6-word sentence:

"The the the the the the."

→ Independence assumption does not match sequential structure of language.

DATOS SECUENCIALES

More realistic model:

Assume conditional dependence of words

$$p(x_T) = p(x_T | x_1, \dots, x_{T-1})$$

Modeling word probabilities is really

?

Context

Target

$p(x|\text{context})$

difficult

0.01

hard

0.009

fun

0.005

...

...

easy

0.00001

DATOS SECUENCIALES

The chain rule

Computing the joint $p(\mathbf{x})$ from conditionals

$$p(\mathbf{x}) = \prod_{t=1}^T p(x_t | x_1, \dots, x_{t-1})$$

Modeling

Modeling word

Modeling word probabilities

Modeling word probabilities is

Modeling word probabilities is really

Modeling word probabilities is really difficult

$$p(x_1)$$

$$p(x_2 | x_1)$$

$$p(x_3 | x_2, x_1)$$

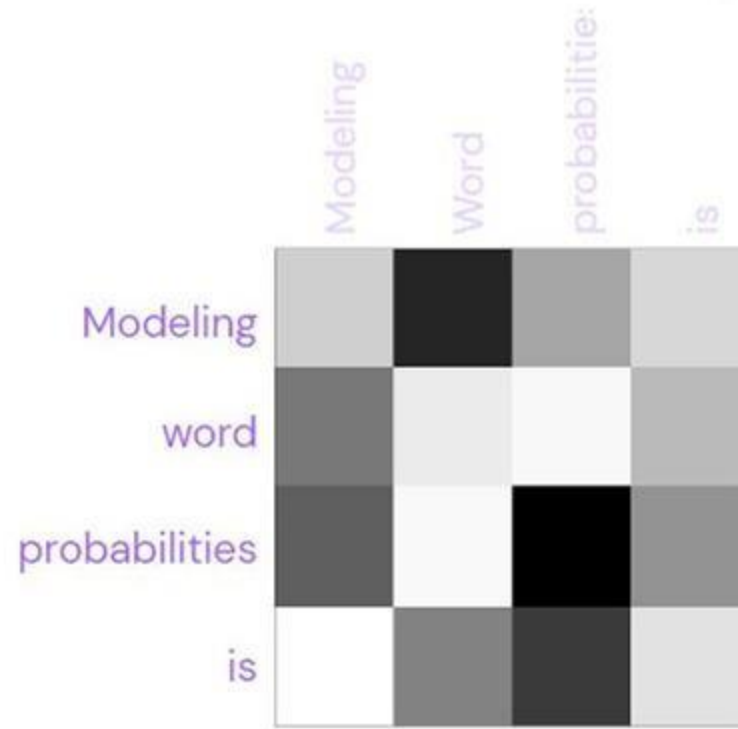
$$p(x_4 | x_3, x_2, x_1)$$

$$p(x_5 | x_4, x_3, x_2, x_1)$$

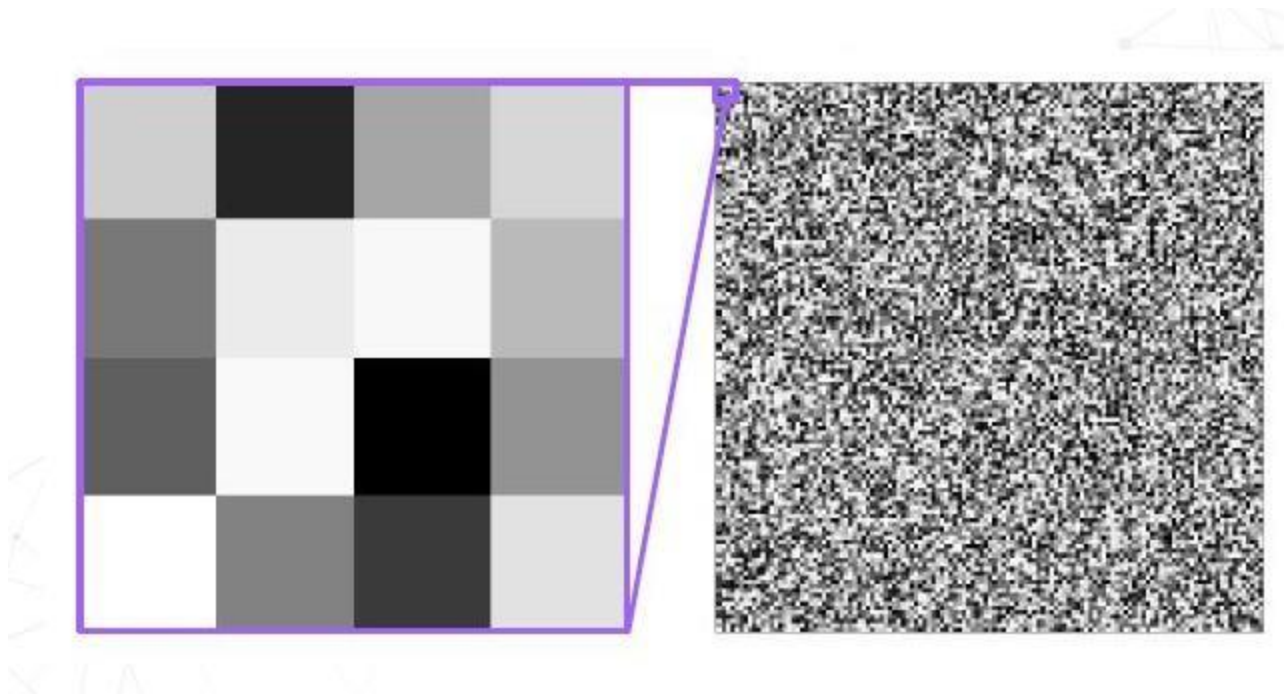
$$p(x_6 | x_5, x_4, x_3, x_2, x_1)$$

PROBLEMA DE ESCALAMIENTO

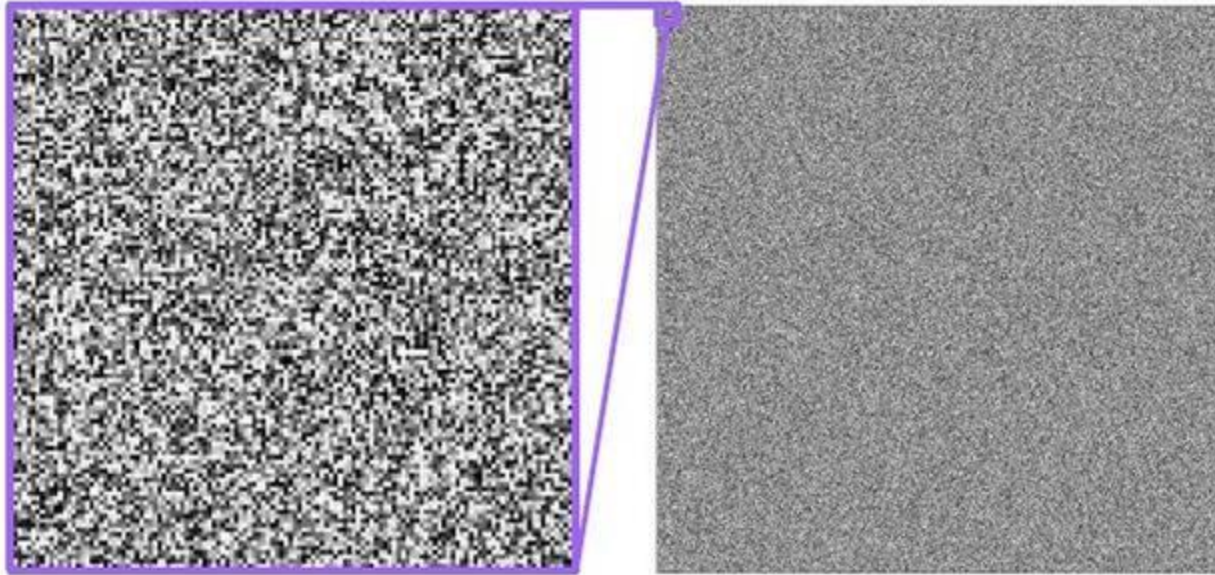
$$p(x_2|x_1)$$



PROBLEMA DE ESCALAMIENTO



Escalamiento



These images are only for context of size $N=1$!
The table size of larger contexts will grow with **vocabulary^N**

N-gramas

Only condition on N previous words

$$p(\mathbf{x}) \approx \prod_{t=1}^T p(x_t | x_{t-N-1}, \dots, x_{t-1})$$

Modeling

Modeling word

Modeling word probabilities

word probabilities is

probabilities is really

is really difficult

$$p(x_1)$$

$$p(x_2 | x_1)$$

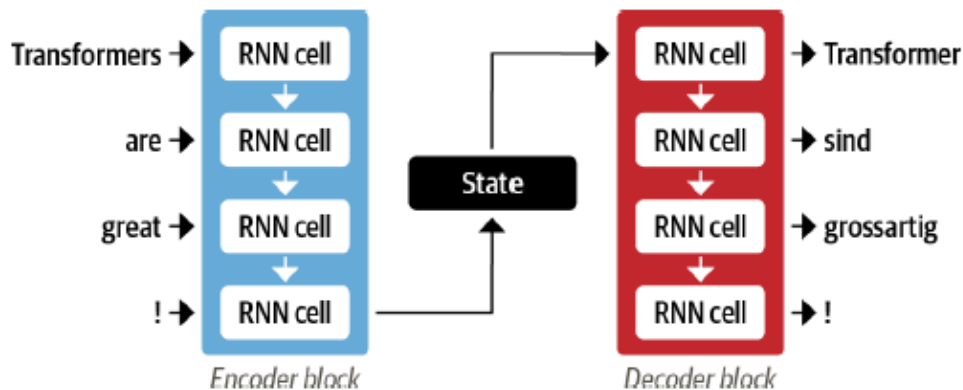
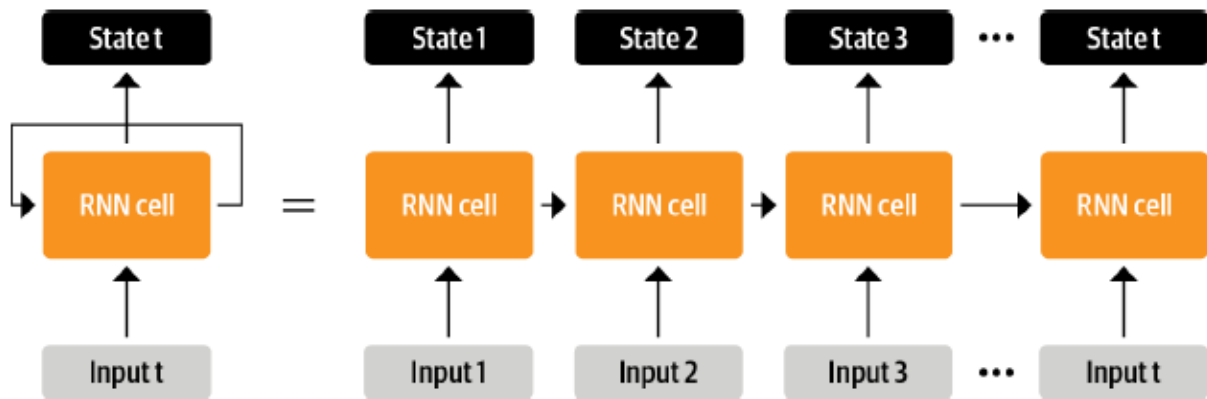
$$p(x_3 | x_2, x_1)$$

$$p(x_4 | x_3, x_2)$$

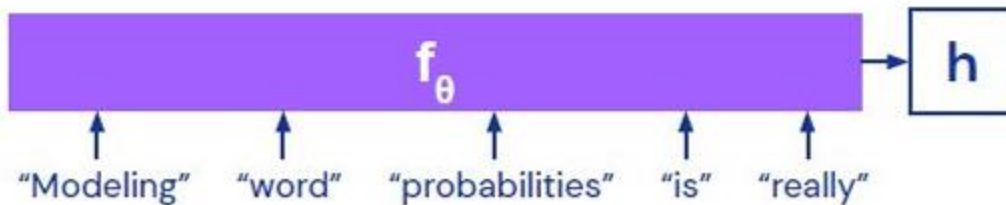
$$p(x_5 | x_4, x_3)$$

$$p(x_6 | x_5, x_4)$$

RNN



RNN

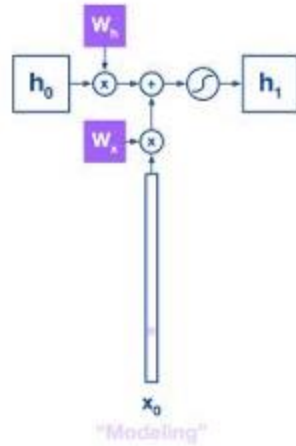


f_{θ} summarises the context in h such that:

$$p(x_t | x_1, \dots, x_{t-1}) \approx p(x_t | h)$$

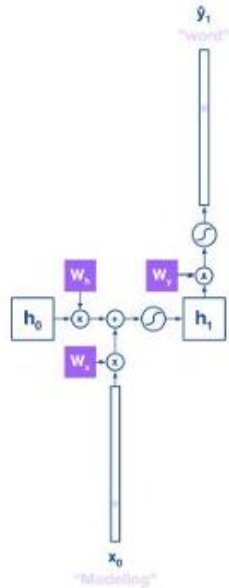


RNN



$$\mathbf{h}_t = \tanh(\mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{W}_x \mathbf{x}_t)$$

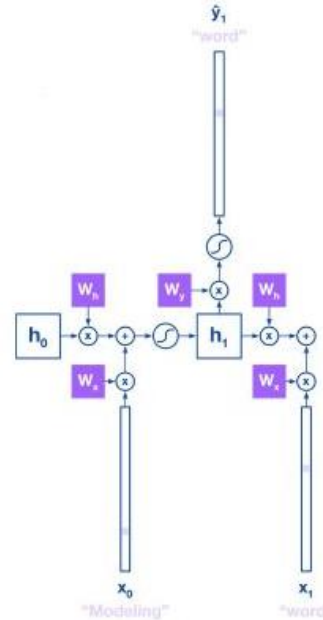
RNN



RNNs predict the target y (the next word) from the state h .

$$p(y_{t+1}) = \text{softmax}(W_y h_t)$$

Softmax ensures we obtain a distribution over all possible words.



Input next word in sentence x_1

Hello Transformers

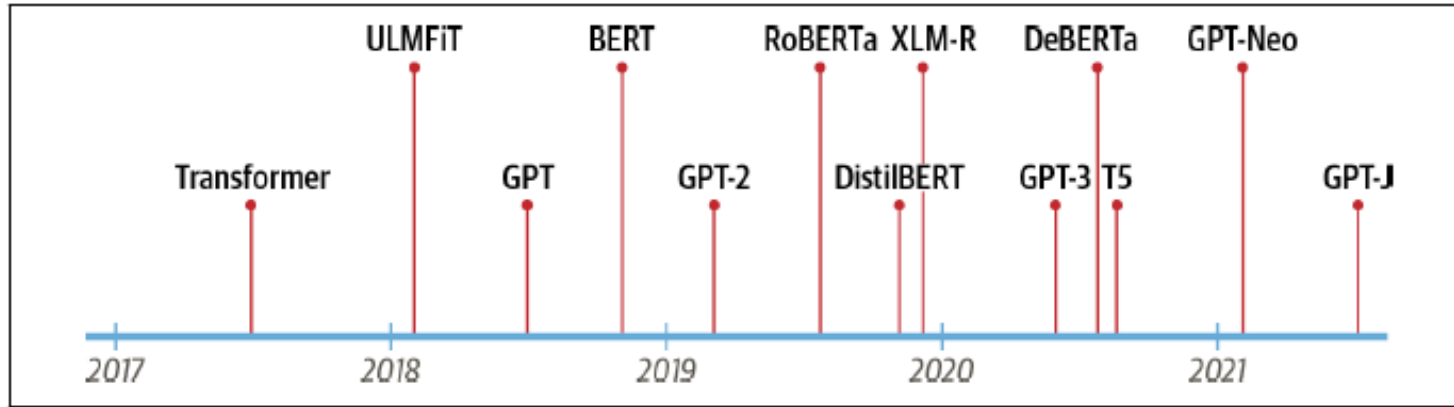
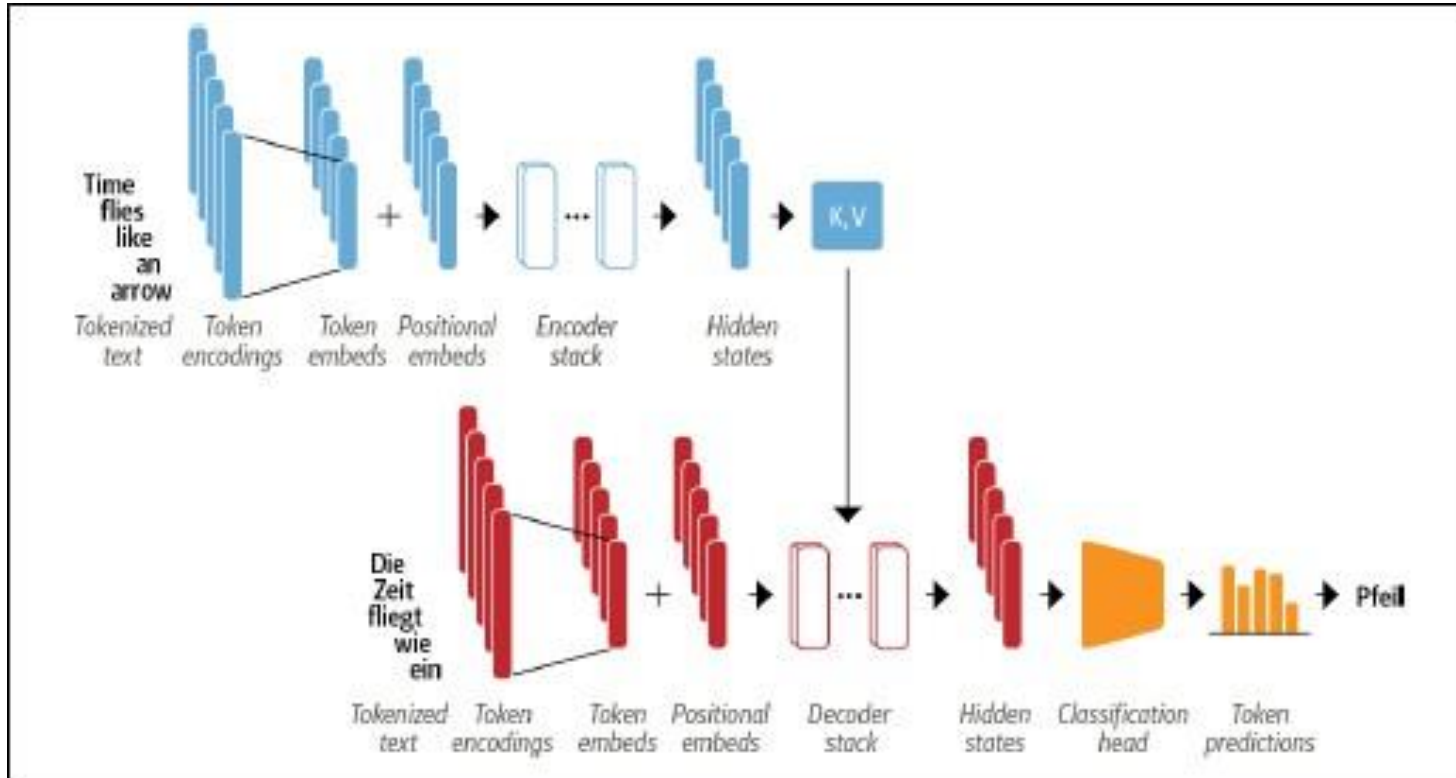
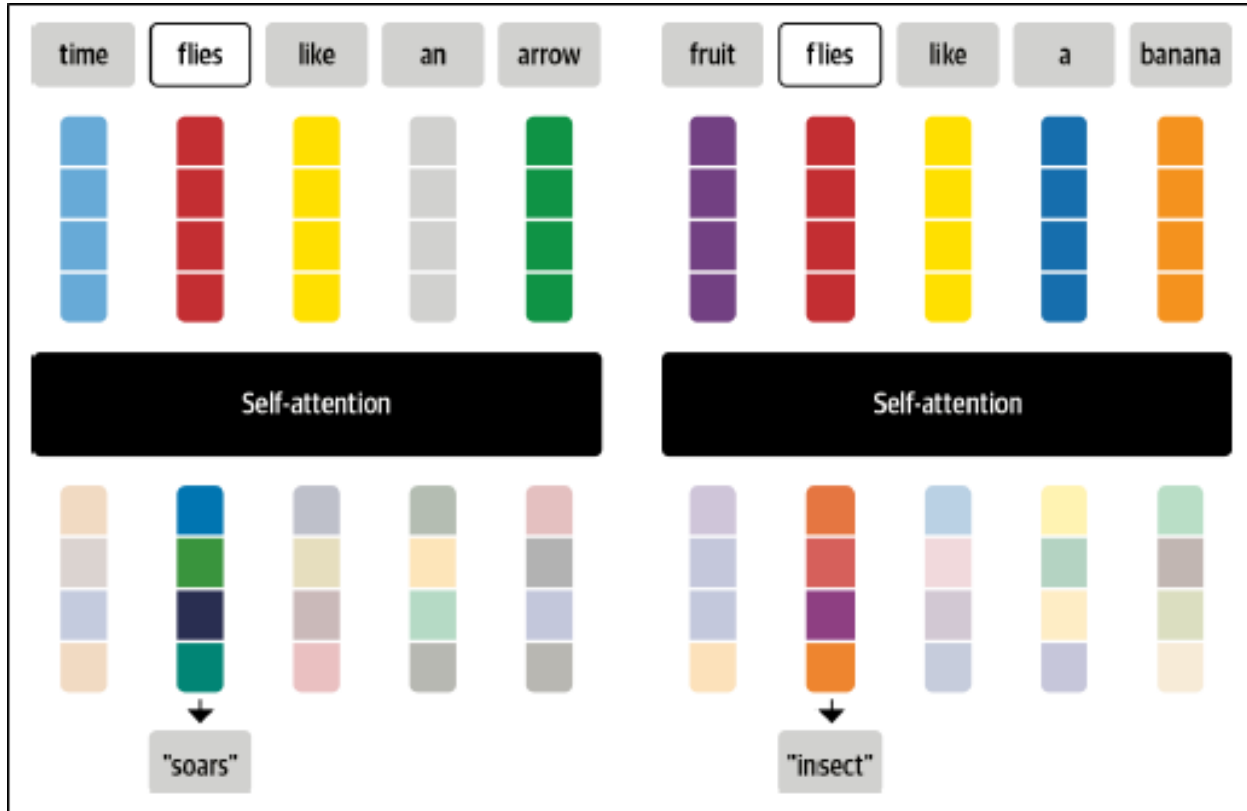


Figure 1-1. The transformers timeline

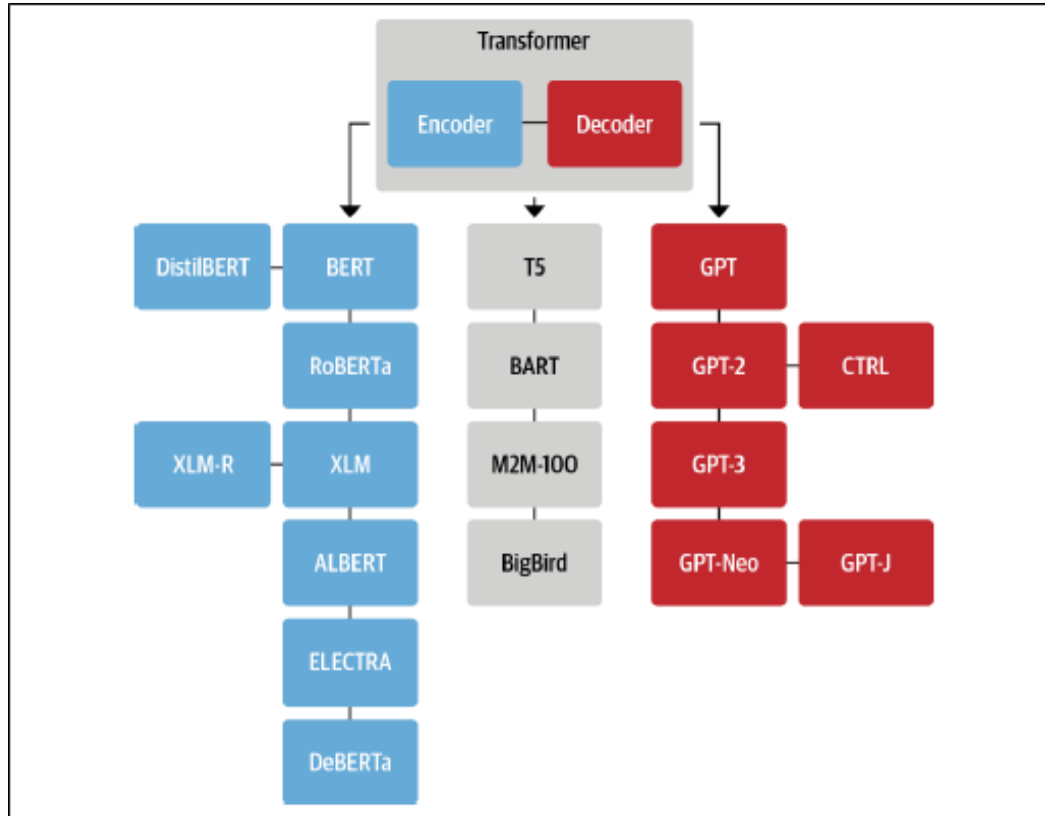
Transformer Architecture



Self-Attention



Algunas Arquitecturas de Transformers



Hugging Face Hub



Hugging Face

Search models, datasets, users...

Models

Datasets

Spaces

Docs

Solutions

Pricing



Log In

Sign Up

Tasks Libraries Datasets Languages Licenses Other

Filter Tasks by name

Multimodal

- Feature Extraction
- Text-to-Image
- Image-to-Text
- Text-to-Video
- Visual Question Answering
- Document Question Answering
- Graph Machine Learning

Computer Vision

- Depth Estimation
- Image Classification
- Object Detection
- Image Segmentation
- Image-to-Image
- Unconditional Image Generation
- Video Classification
- Zero-Shot Image Classification

Natural Language Processing

- Text Classification
- Token Classification
- Table Question Answering
- Question Answering
- Zero-Shot Classification
- Translation
- Summarization
- Conversational
- Text Generation
- Text2Text Generation
- Fill-Mask
- Sentence Similarity

Models 372,969

Filter by name

new Full-text search

Sort: Trending

adept/fuyu-8b

Text Generation • Updated 4 days ago • \pm 14.2k • \heartsuit 508

segmind/SSD-1B

Text-to-Image • Updated about 8 hours ago • \pm 8.62k • \heartsuit 175

mistralai/Mistral-7B-v0.1

Text Generation • Updated 14 days ago • \pm 277k • \heartsuit 1.48k

stabilityai/stable-diffusion-xl-base-1.0

Text-to-Image • Updated 24 days ago • \pm 7.37M • \heartsuit 3.28k

meta-llama/Llama-2-7b-chat-hf

Text Generation • Updated 4 days ago • \pm 975k • \heartsuit 1.55k

CausalLM/7B

Text Generation • Updated 1 day ago • \pm 184 • \heartsuit 74

meta-llama/Llama-2-7b

Text Generation • Updated Jul 19 • \heartsuit 2.86k

SimianLuo/LCM_Dreamshaper_v7

Text-to-Image • Updated 1 day ago • \pm 24.1k • \heartsuit 78

jinaai/jina-embeddings-v2-base-en

Feature Extraction • Updated about 3 hours ago • \pm 4.17k • \heartsuit 183

HuggingFaceH4/zephyr-7b-alpha

Text Generation • Updated about 11 hours ago • \pm 58.6k • \heartsuit 799

CausalLM/14B

Text Generation • Updated about 5 hours ago • \pm 323 • \heartsuit 136

amazon/MistralLite

Text Generation • Updated 3 days ago • \pm 2.94k • \heartsuit 122

mistralai/Mistral-7B-Instruct-v0.1

Text Generation • Updated 15 days ago • \pm 220k • \heartsuit 864

teknium/OpenHezmes-2-Mistral-7B

Text Generation • Updated about 19 hours ago • \pm 10.1k • \heartsuit 143

SkunkworksAI/BakLLaVA-1

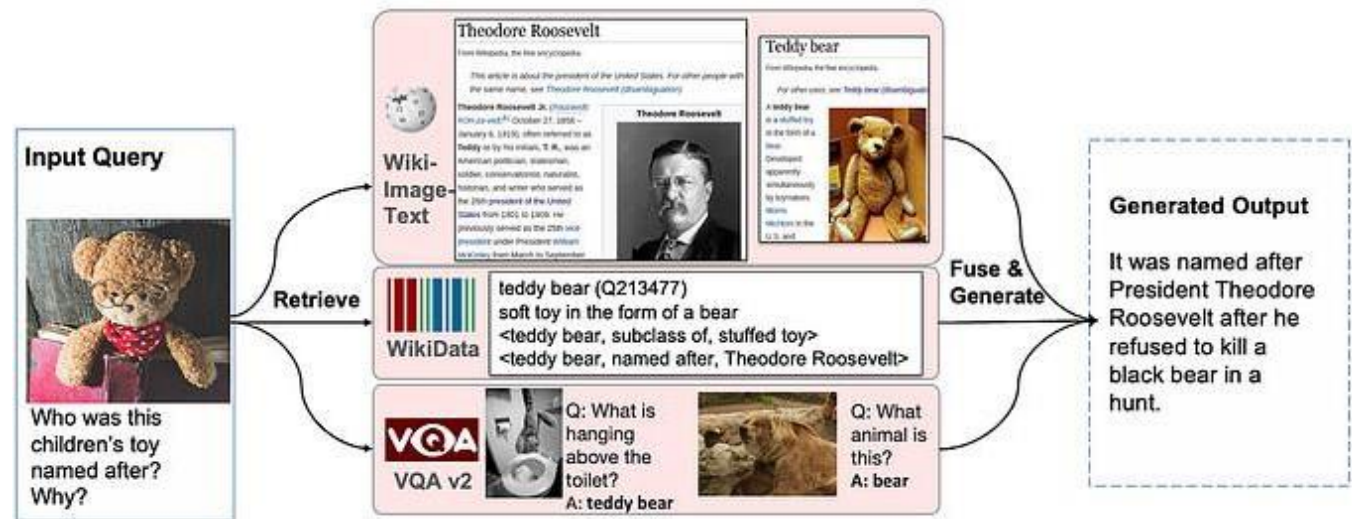
Text Generation • Updated 3 days ago • \pm 523 • \heartsuit 175

runwayml/stable-diffusion-v1-5

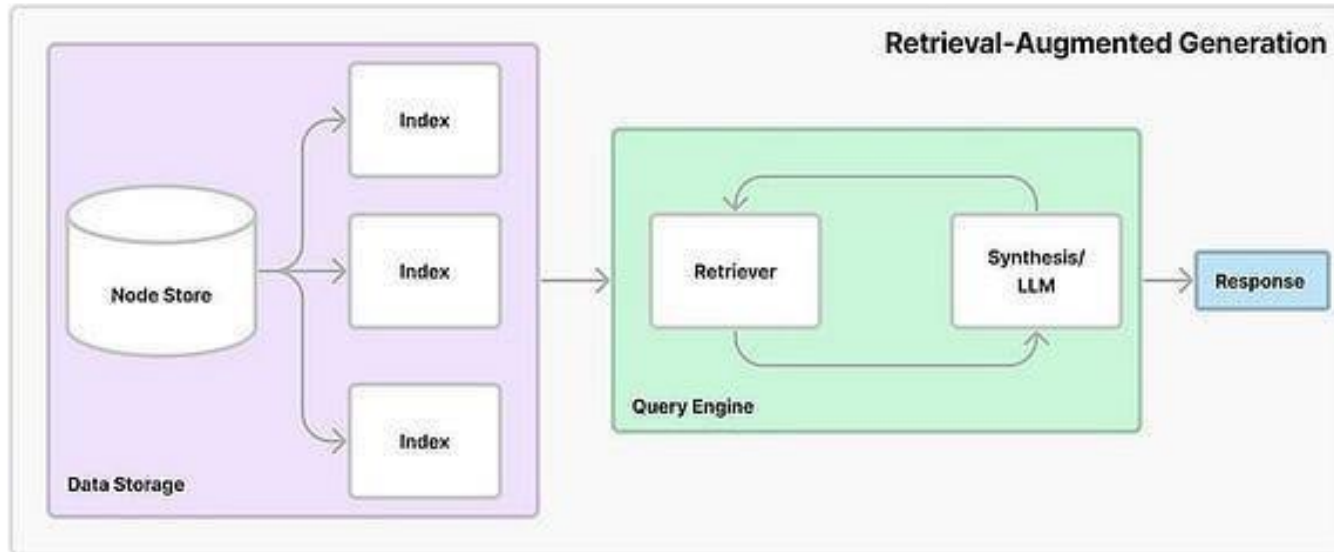
Text-to-Image • Updated Aug 23 • \pm 7.65M • \heartsuit 9.47k

Retrieval-Augmented Generation (RAG)

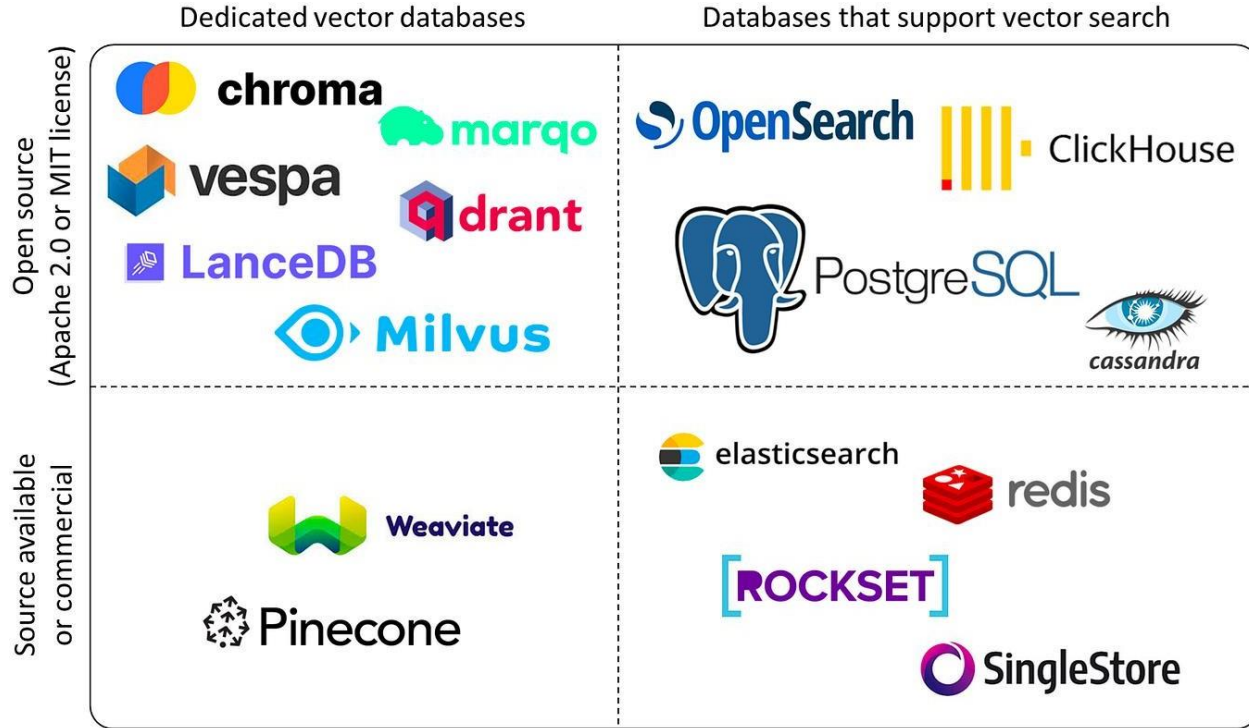
Retrieval-augmented generation (RAG) is a natural language processing (NLP) approach that combines the benefits of both retrieval-based and generation-based methods for content generation tasks. It aims to improve the quality and controllability of the generation tasks by leveraging a pre-trained language model in conjunction with a retrieval mechanism.



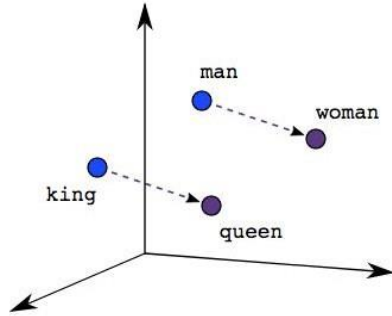
RAG: Componentes



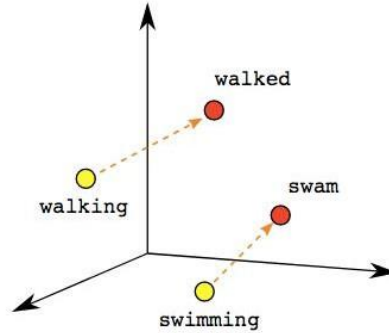
RAG: Database



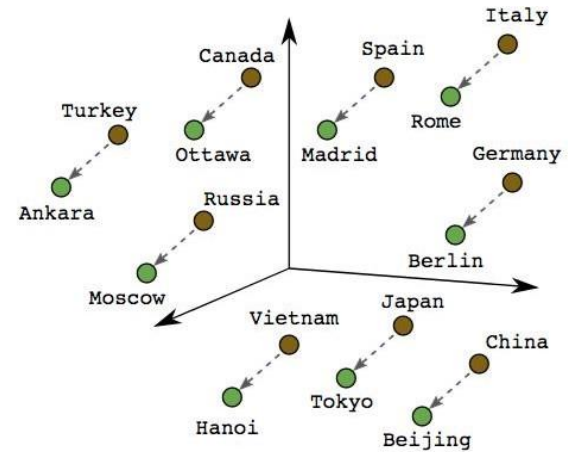
RAG: Embeddings



Male-Female



Verb Tense



Country-Capital



Hugging Face

The task illustrated in this tutorial is supported by the following model architectures:

ALBERT, BART, BERT, BigBird, BigBird-Pegasus, BLOOM, CamemBERT, CANINE, ConvBERT, Data2VecText, DeBERTa, DeBERTa-v2, DistilBERT, ELECTRA, ERNIE, ErnieM, Falcon, FlauBERT, FNet, Funnel Transformer, OpenAI GPT-2, GPT Neo, GPT NeoX, GPT-J, I-BERT, LayoutLMv2, LayoutLMv3, LED, LiLT, Longformer, LUKE, LXMERT, MarkupLM, mBART, MEGA, Megatron-BERT, MobileBERT, MPNet, MPT, MRA, MT5, MVP, Nezha, Nyströmformer, OPT, QDQBert, Reformer, RemBERT, RoBERTa, RoBERTa-PreLayerNorm, RoCBert, RoFormer, Splinter, SqueezeBERT, T5, UMT5, XLNet, RoBERTa-XL, XLNet, X-MOD, YOSO