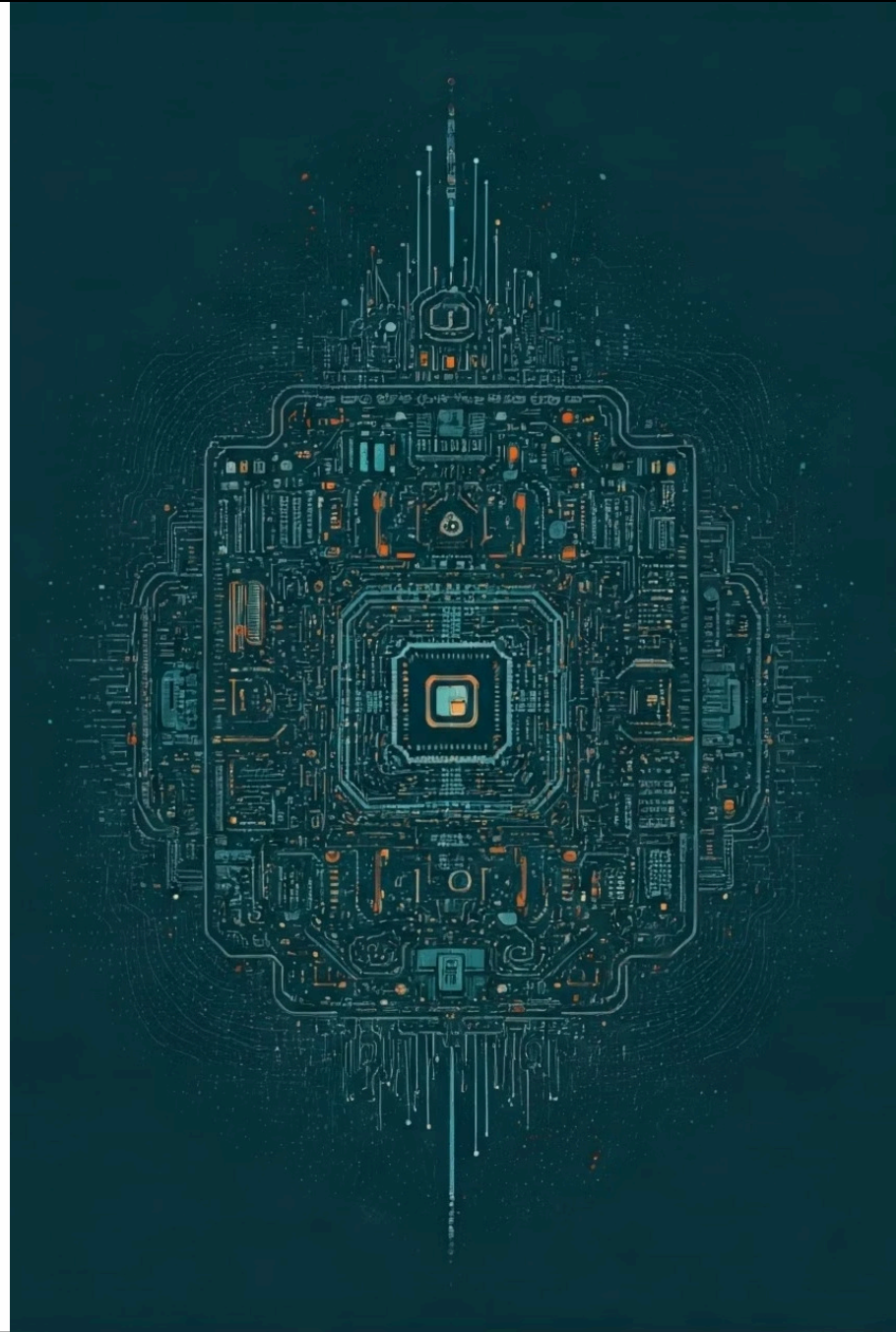
Language Models

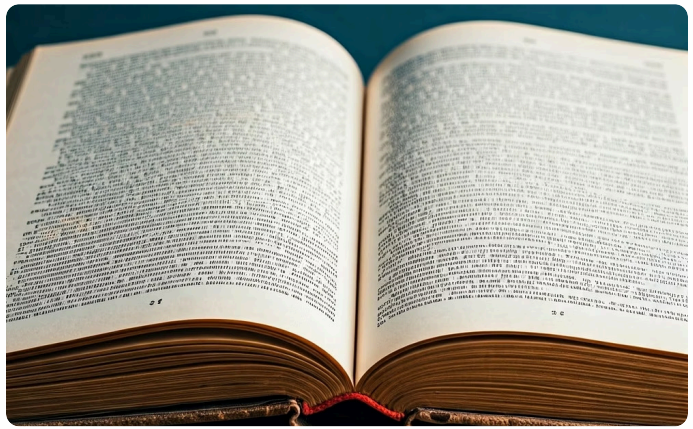
Andrés Muñoz / Iván Ruiz



Contents

Anatomy of Language Models

Language Models



Natural Language Processing

Designed to understand, generate and manipulate text in natural language.

Machine Learning

These are **models** of AI trained on large datasets of text, learning the rules, patterns and structures of language.

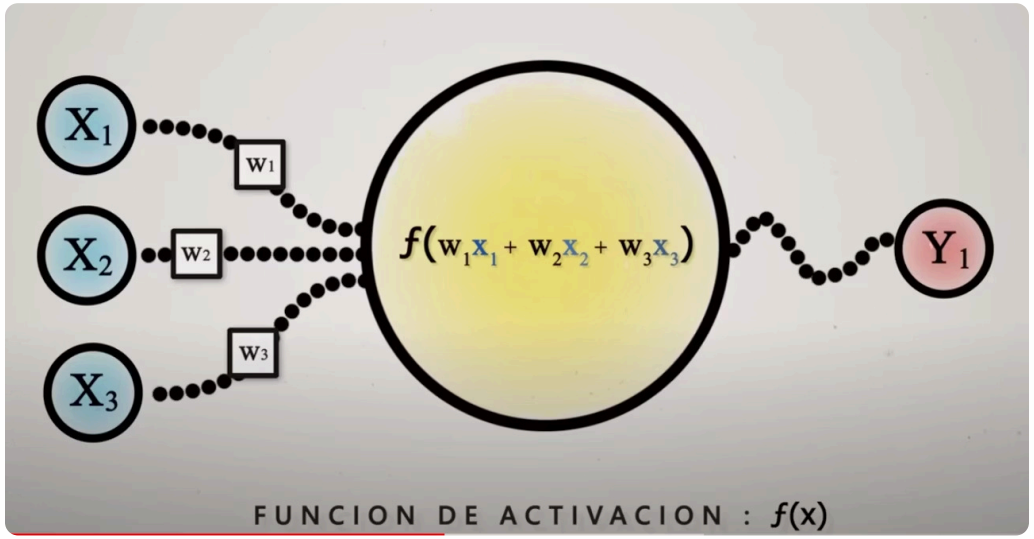
Artificial Neural Networks

Algorithms inspired by the functioning of the human brain, designed to recognise patterns and process data efficiently.

But first...Artificial Neuron!

It is a processing node that receives inputs, performs a mathematical operation on them, and produces an output.

The mathematical operation, called the activation function, determines the behaviour of the neuron.



See how an Artificial Neuron Network works: ghttps://www.youtube.com/watch?v=jmmW0F0biz0

Neural Network Architectures

Multilayer Perceptron (MLP)

MLPs are neural networks that use hidden layers to learn complex functions. They are useful for pattern recognition, such as images or text.

Convolutional Neural Network (CNN)

CNNs are specialised neural networks for image analysis. They employ convolution operations to extract relevant features, such as edges and textures.

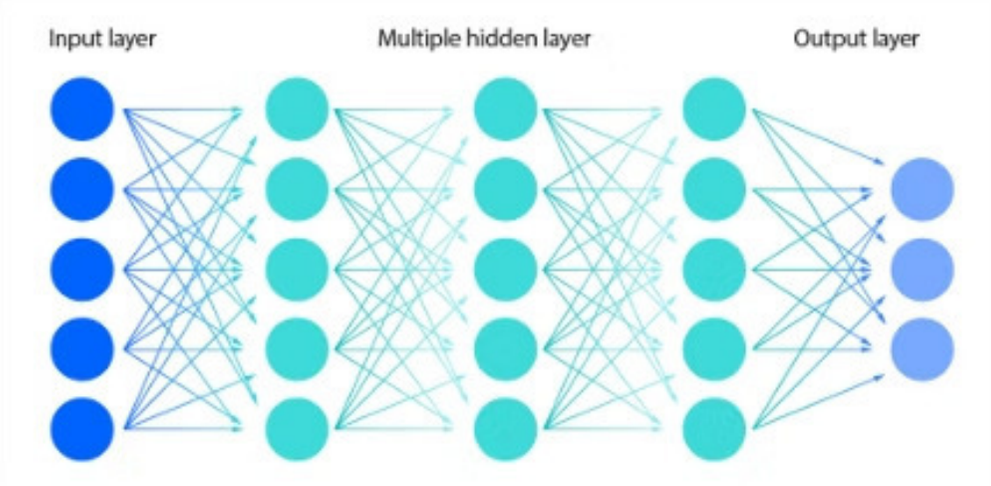
Recurrent Neural Network (RNN)

RNNs process data sequences, such as natural language or speech recognition. They allow "remembering" previous information in the sequence.

Generative Adversarial Network (GAN)

GANs consist of two neural networks that compete to generate realistic data. A generator creates data and a discriminator distinguishes it.

The evolution: Deep Learning



Input layer

Receives the input data, such as text or images.

Hidden layers

Process the information and extract relevant features at multiple levels.

Output layer

Generates the final output, such as a prediction or a classification.

https://www.ibm.com/en-gb/topics/neural-networks

Learn the difference!! Machine Learning vs Deep Learning

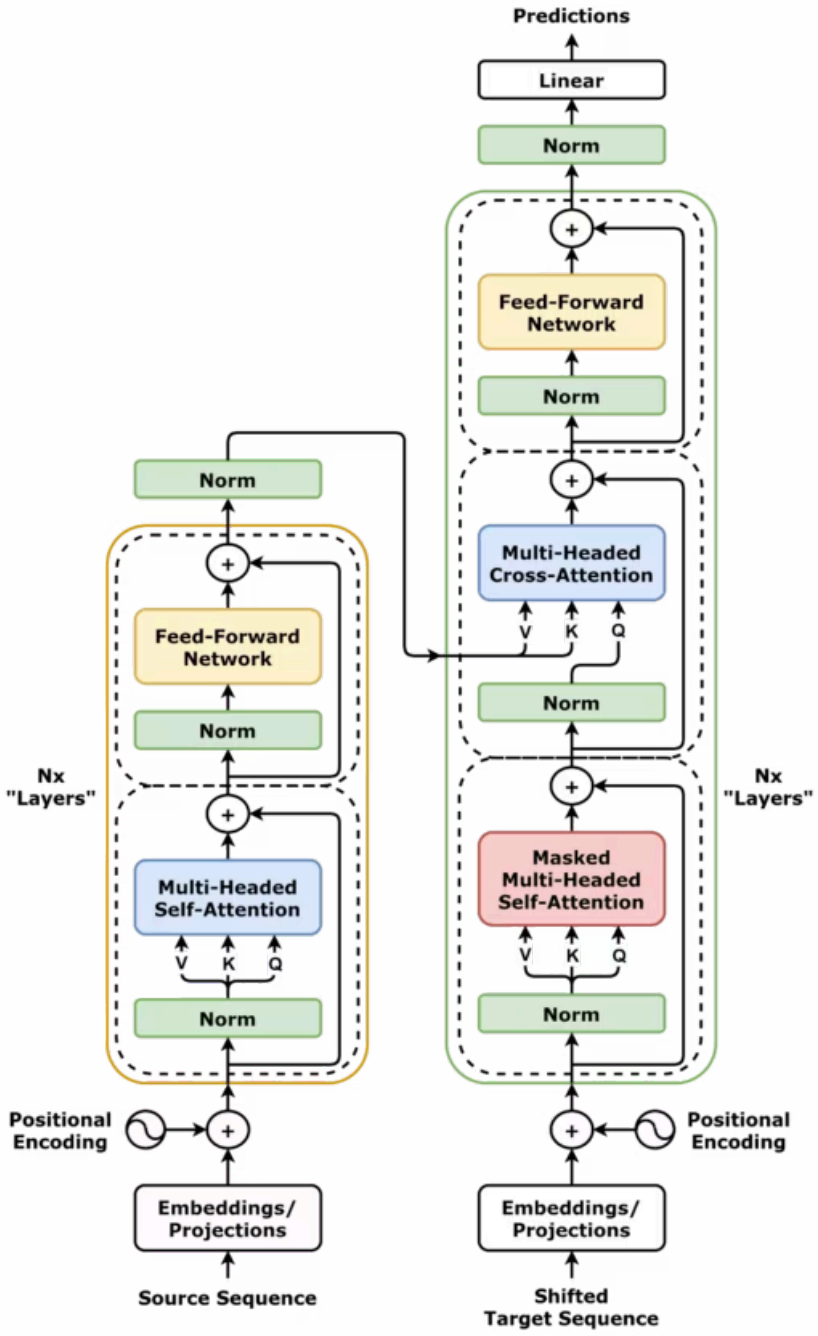
Finally...Architecture of Language Models

Transformers

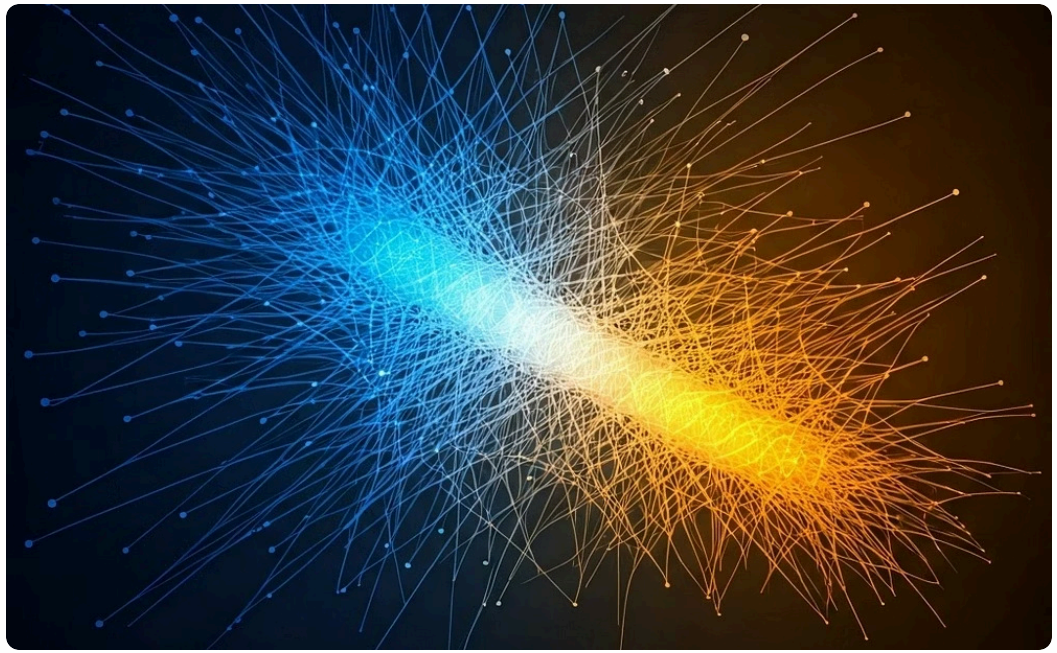
Language models leverage the *Transformer* architecture, which allows them to capture contextual relationships efficiently and significantly improve natural language understanding.

GPT = Generative Pretrained Transformer

[Transformer Explainer: LLM Transformer Model Visually Explained]



Key Processes



Training

Language models are trained using neural networks that adjust the weights of connections between neurons. This training requires a large amount of data.

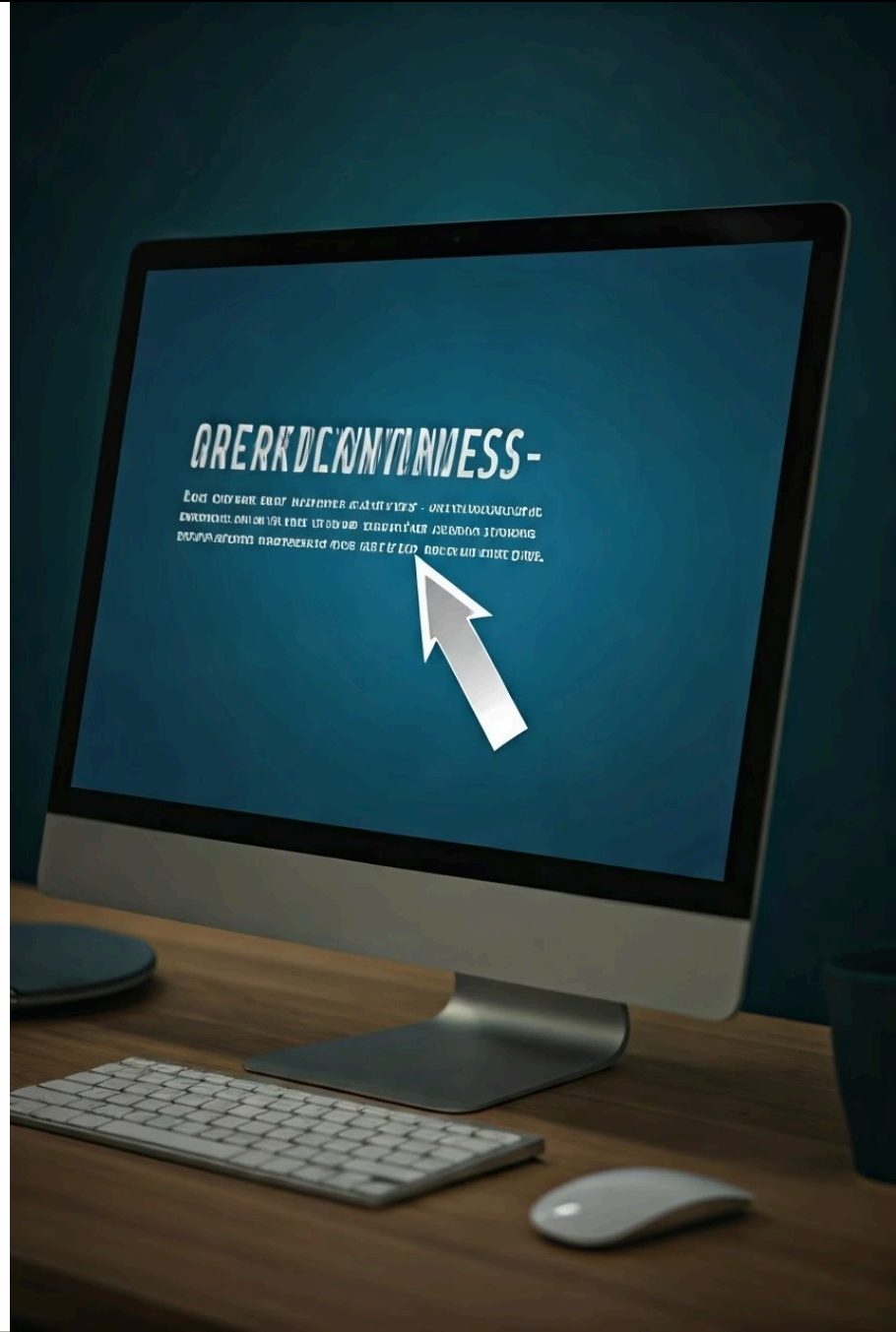
Inference

During inference, language models use auto-regressive generation to predict the next word in a sequence. The process ends when a maximum length is reached or a special token is generated.

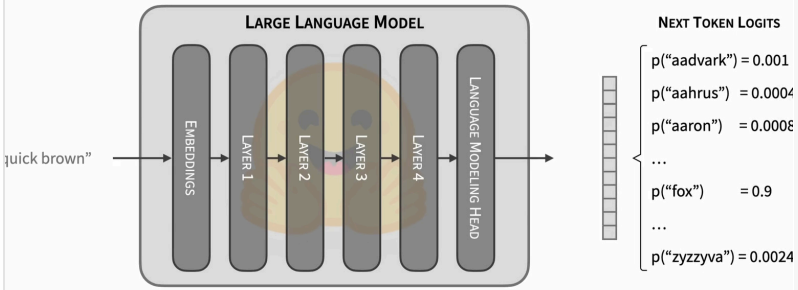
Tokens

A token is the basic unit of text that language models use to process and generate language.

Number of Tokens≈Number of Words×1.3



Basic operation



huggingface

Generation with LLMs

We're on a journey to advance and democratize artificial intelligence through open source and open science.

Let's play with a tokenizer! https://huggingface.co/spaces/Xenova/the-tokenizer-playground

Embeddings

Embeddings are **mathematical representations** of the meaning of words or phrases.

Each word or phrase is converted into a **numerical vector**, a set of numbers that reflects its semantic meaning.

These representations allow **mathematical operations** to be performed to compare the meaning of words or phrases.

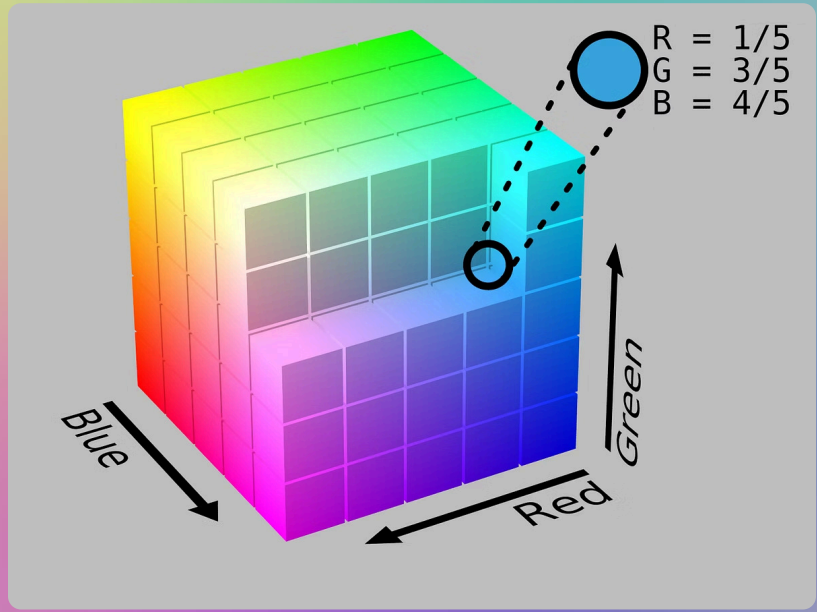
Embeddings

Example

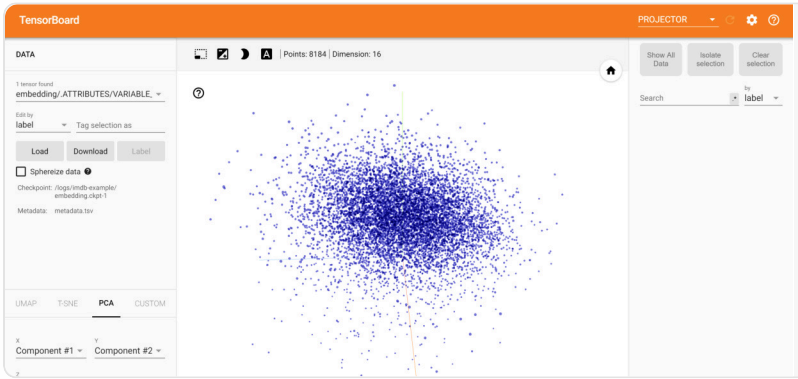
Yellow -> (255, 255, 0)
Green -> (0, 255, 0)
Blue -> (0, 0, 255)

Mathematical operations

Yellow – Green + Blue = Magenta
(255, 255, 0) - (0, 255, 0) + (0, 0, 255) = (255, 0, 255)



Visualisation of embeddings

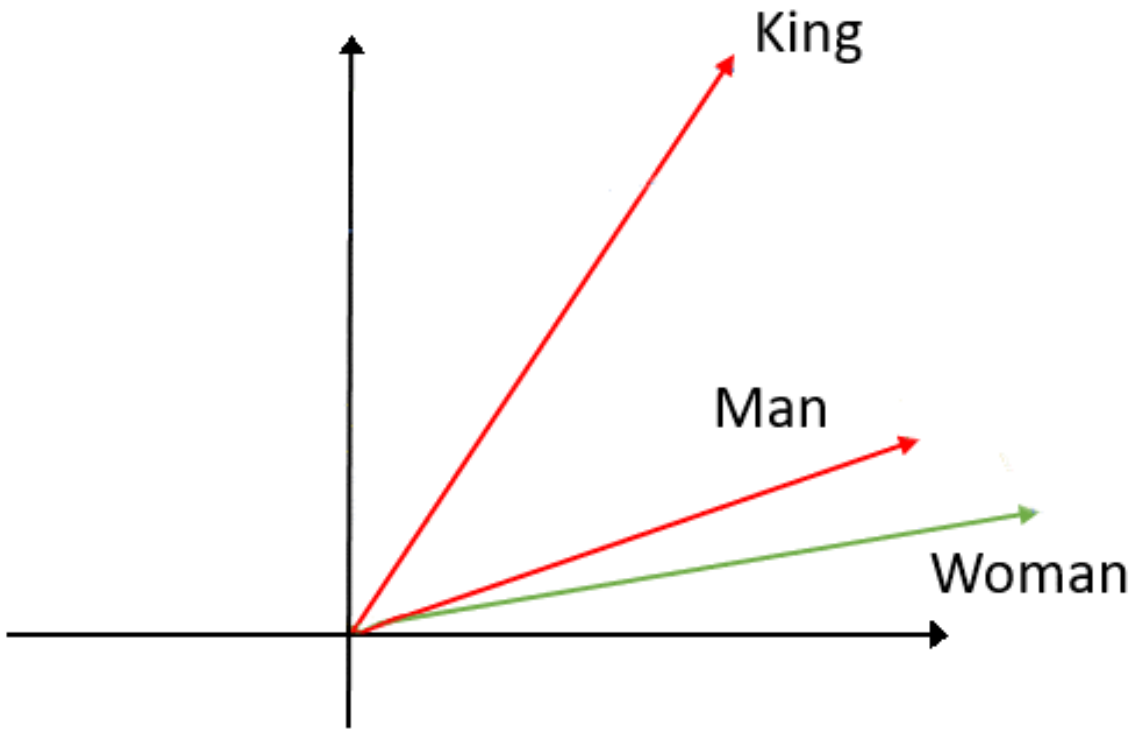


projector.tensorflow.org

Embedding projector – visualization of high-dimensional data

Visualize high dimensional data.

King - Man + Woman = Queen



Try yours: [WebVectors: Semantic Calculator], e.g. Doctor - Hospital + School = ?? // Bird - Fly + Swim = ??

Stages in Building a Model

1

Model Architecture and Size

The first step involves defining the model's architecture and size. This includes choosing the appropriate neural network structure and determining the number of parameters and layers.

2

Data Collection and Cleaning

Data is collected from various sources, including public Internet resources and external data providers. The data is then cleaned and preprocessed to ensure quality and suitability for training.

3

Unsupervised Pre-training

Pre-training allows the model to learn the underlying linguistic representations without explicit supervision, in order to understand language and generate coherent text.

4

Fine-tuning with Labelled Data

The model is fine-tuned with labelled data specific to the desired task, such as conversing, through question-answer pairs.

5

Reinforcement Learning with Human Feedback (RLHF)

Human feedback is incorporated into the training process to improve the model's performance. This involves adjusting the model's responses based on evaluations made by humans.

6

Evaluation

The model is evaluated to assess its accuracy, coherence, relevance, fluency, and generalization capability. Metrics are used to measure the model's performance and ensure its quality.

LLM as a Black Box

LLM as a black box

1

Text input

The user introduces the text they want the LLM to process. Language models process the text as a sequence of **tokens**.

2

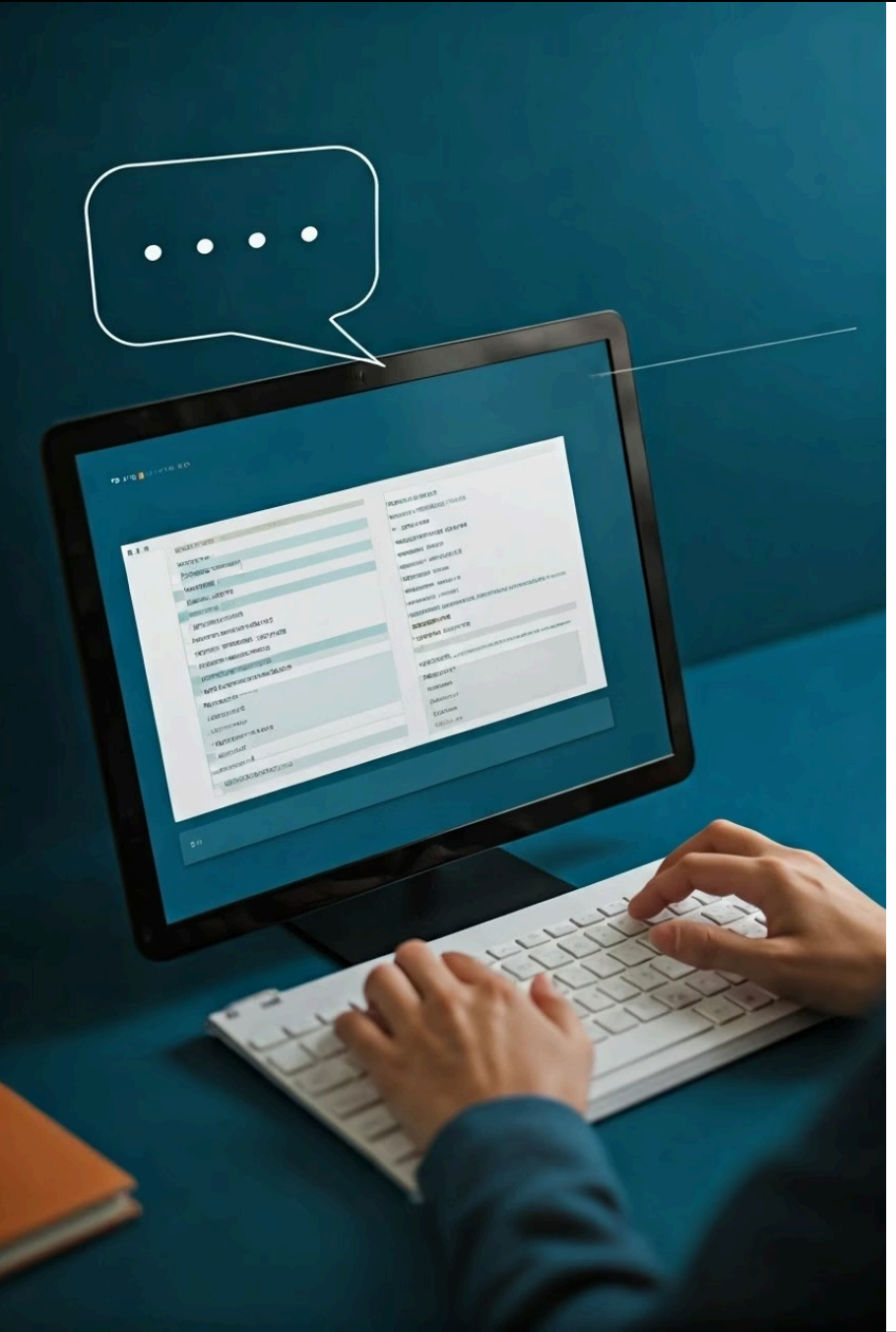
LLM inference engine

The inference engine uses the language model to process the input text, converting the tokens into embeddings and calculating the output probabilities for each token.

3

Text output

The language model generates text as output, based on the prediction of tokens, creating a coherent and meaningful text.



Text input/output

Input

A **prompt** is an instruction or question given to the language model.

Output

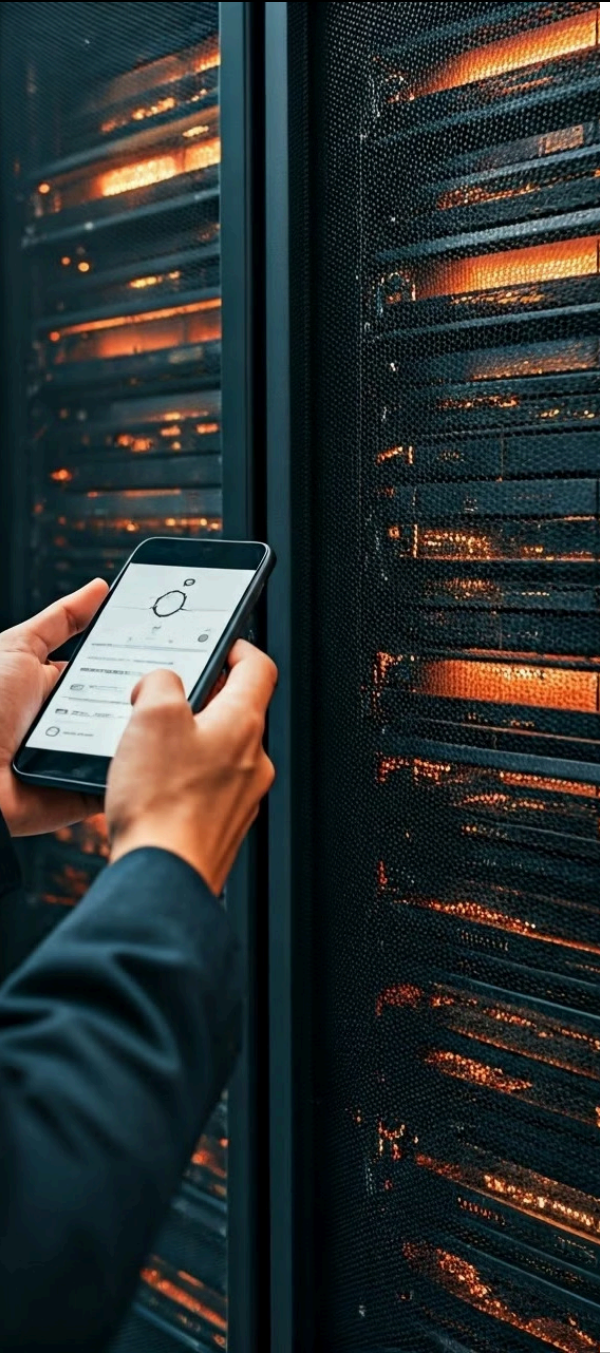
The model generates a sequence of coherent words and phrases based on the input.

Quality

The quality of the generated response depends on the design of the prompt.

Precision

It is important to design prompts with precision and consistency.



Inference Engine

Model Execution

Execute pre-trained language models.

Input and Output

Process new inputs to generate relevant outputs.

Scalability

Handle multiple requests simultaneously.

REST API

Facilitate integration within applications.

Model Formats

Language models are generally stored as binary files containing the **weights** of the model

PyTorch (.pt or .pth)

PyTorch is known for its flexibility and ease of use.

TensorFlow (ckpt and SavedModel)

TensorFlow is an open-source machine learning platform.

Keras (.h5)

Keras, the high-level API of TensorFlow, makes rapid prototyping easier.

Open Neural Network Exchange (.onnx)

ONNX is an open machine learning model format that enables interoperability between different frameworks.

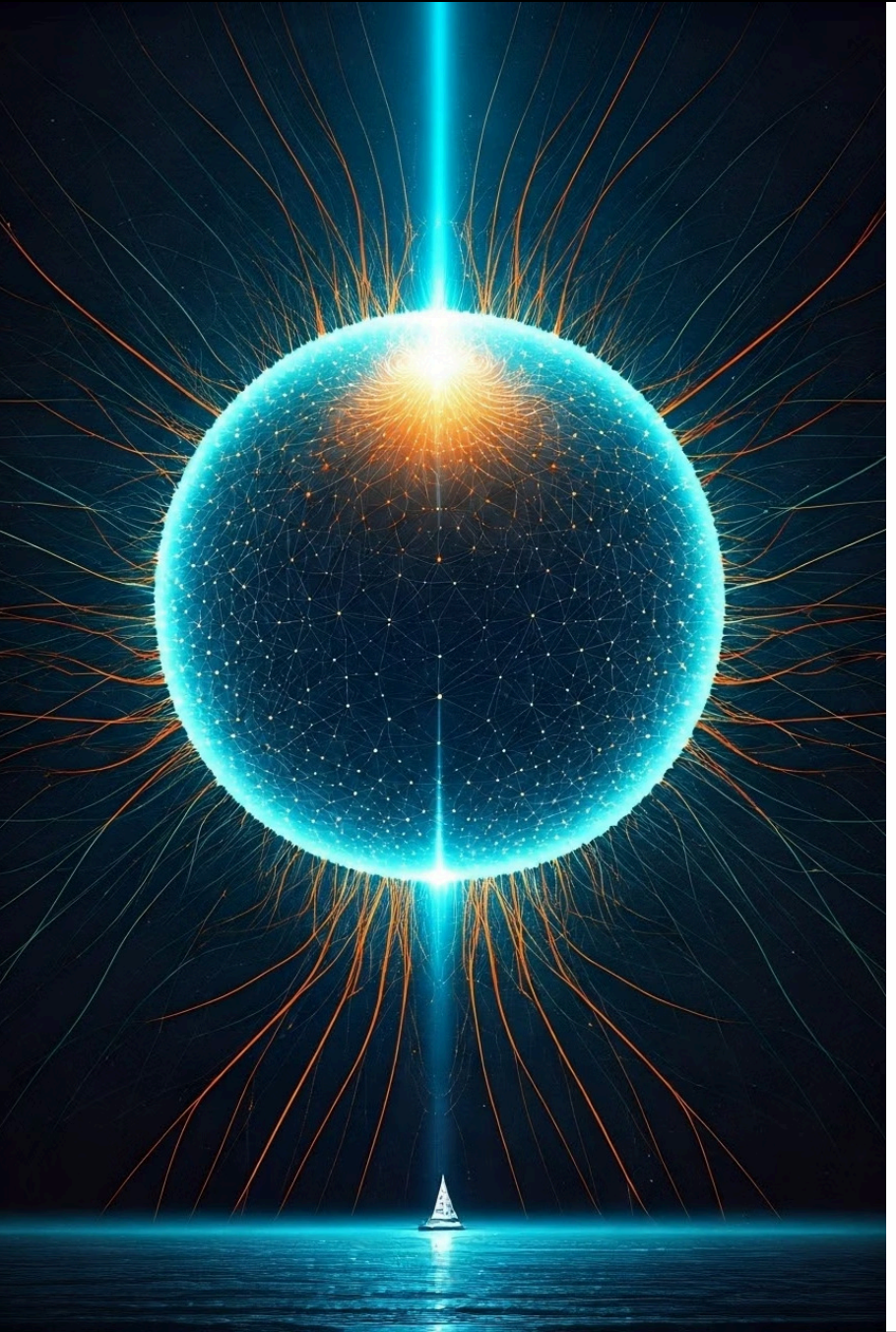
GGML/GGUF

Optimise storage and inference on resource-constrained devices.

Hugging Face Transformers (.bin)

Hugging Face Transformers provides a library of pre-trained models.

Types of Language Models



Foundational models

Broad Capability

Trained on vast unlabelled data corpora. Perform a wide range of general tasks.

Significant Investment

Ground-up development requires millions of dollars. Extensive computational infrastructure and resources.

Notable Examples

GPT-4 and Llama3-pretrained stand out as widely recognised foundational models.

Instruct Models

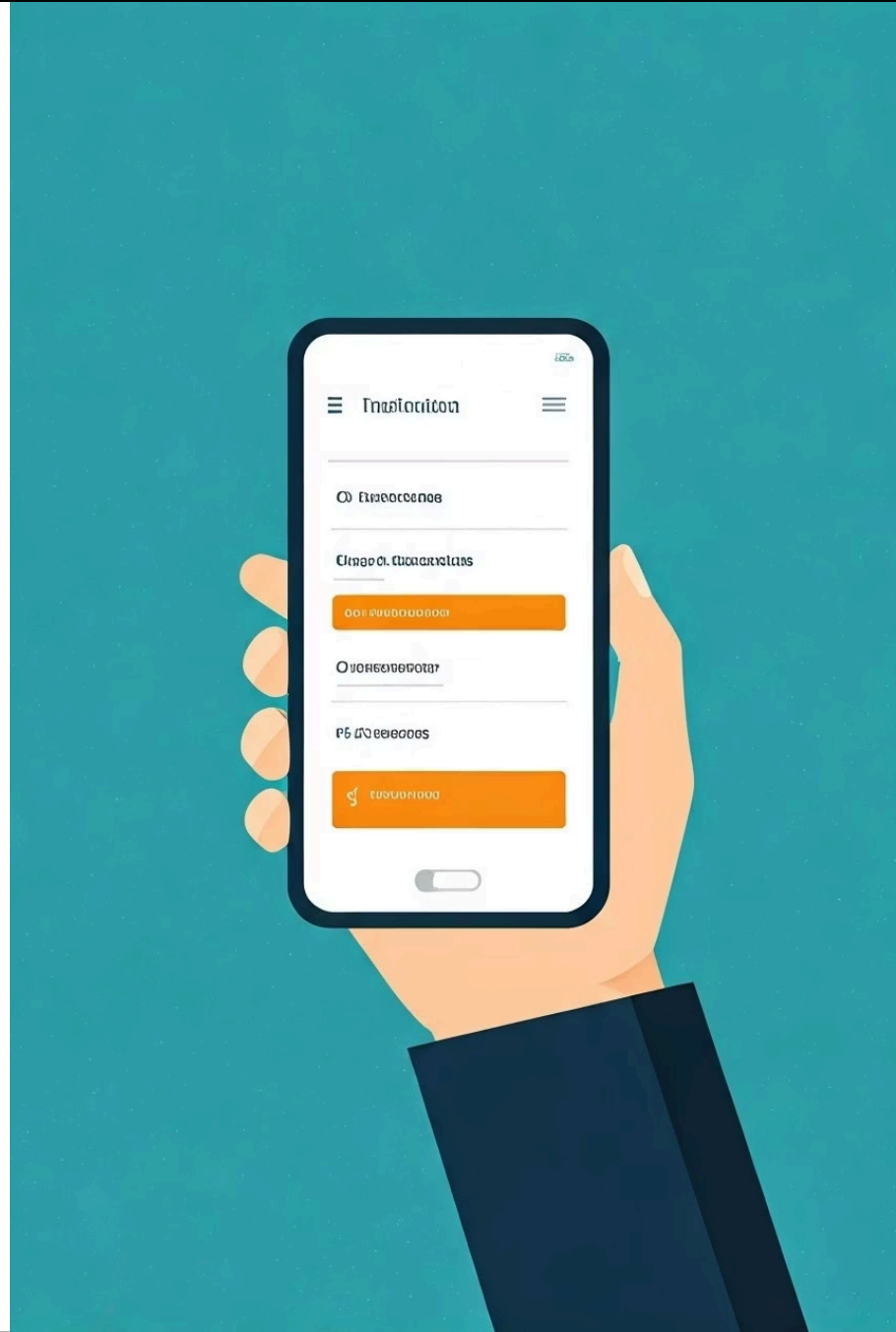
Models that have been trained on datasets that include specific instructions, allowing them to follow directions more precisely and consistently.

These models are particularly useful for interactive applications where the model needs to understand and execute detailed user instructions.

Better ability to follow orders and perform specific tasks.

Responses more aligned with user expectations.

Example: InstructGPT (**deprecated**); gpt-3.5-turbo-instruct (Open AI)

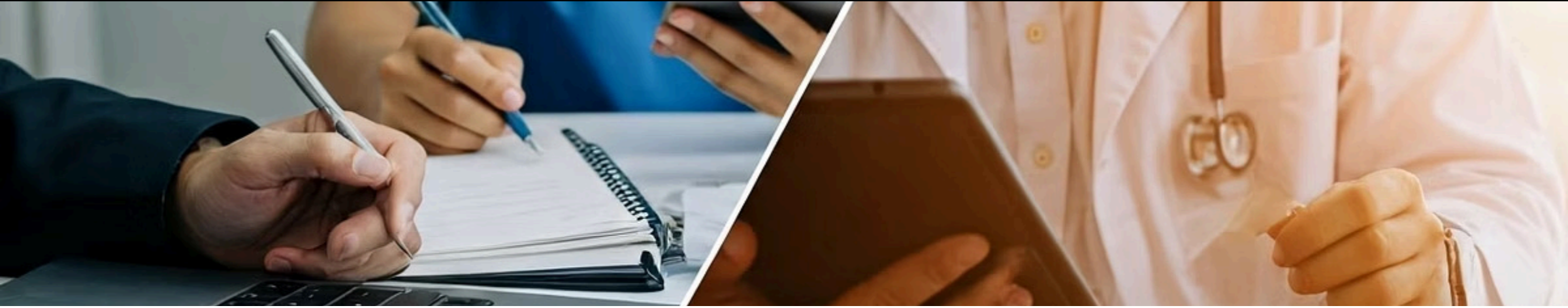


Conversational (Chat) Models

These models are trained to interact with users to maintain fluent and coherent conversations.

They are used in virtual assistants, chatbots and customer support systems.

They are capable of maintaining context, offering natural and logical responses, adapting to the user's style and preferences.



Domain-Specific Models

Specialisation

Models focused on specific fields. They offer greater precision in particular areas.

Advantages

In-depth understanding of specialised terminology and contexts. Improve decision-making in specific sectors.

Examples

LegalGPT for law, ClinicalCamel in medicine, and FinBERT in finance.

Multimodal Models

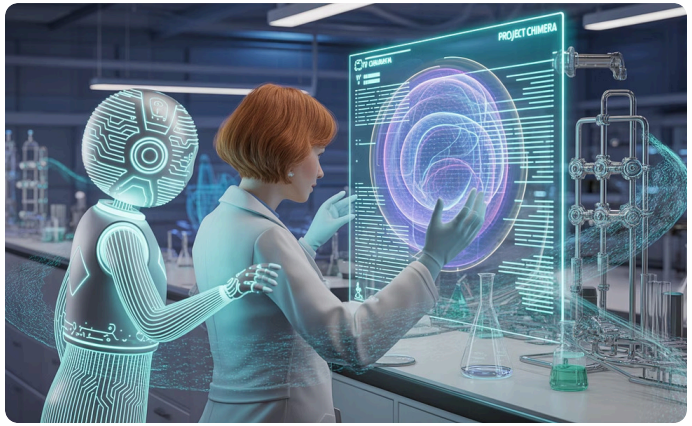
The latest language models not only process text, but can also interact with different data formats.

These models can receive images, audio, video and text as input, and generate responses in any combination of these formats.

Multimodal LLMs represent a significant advancement in artificial intelligence, opening up new possibilities for human-machine interaction.

Examples: GPT-4o, Gemini 2.5 Pro

LLM Reasoners: A New Generation



Explicit Reasoning

Models that go beyond text generation to perform structured reasoning across multiple steps, contexts, or goals.

Key Characteristics

- Trained to reason, plan, and self-evaluate
- Combine symbolic logic with prediction
- Incorporate memory and tool use

Real-World Applications

- Scientific research assistants
- Code debugging agents
- Legal or medical reasoning advisors

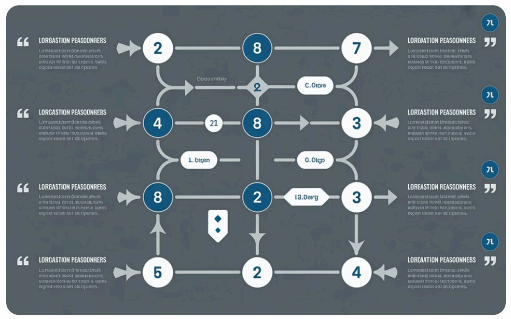
🧠 Example: GPT-4 with chain-of-thought prompting + external tools becomes a reasoner

How LLM Reasoners Differ from Standard LLMs

Standard LLMs	LLM Reasoners
Generate text in response to a prompt	Generate and **analyze** intermediate steps
Typically follow a single instruction	Break down complex queries into subtasks
Rely on prompt clarity	Use internal logic, memory, or planning
Static response	Can **revise, retry, and justify** answers
Minimal control flow	Conditional branches, loops, judgment logic

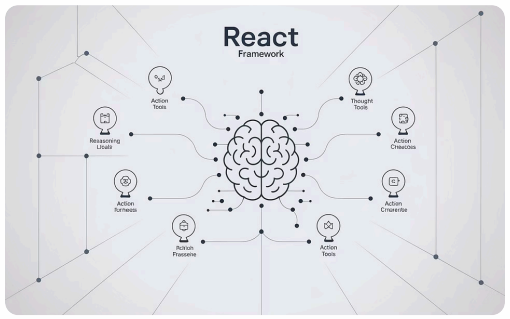
LLM Architectures Enabling Reasoning

Different approaches allow language models to perform structured reasoning:



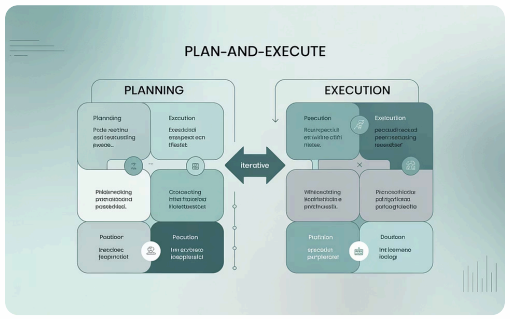
Chain-of-Thought (CoT)

Forces step-by-step logical outputs to make reasoning explicit.



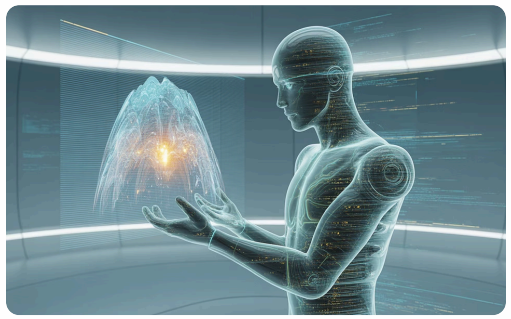
ReAct

Mixes reasoning and acting through strategic tool use.



Plan-and-Execute

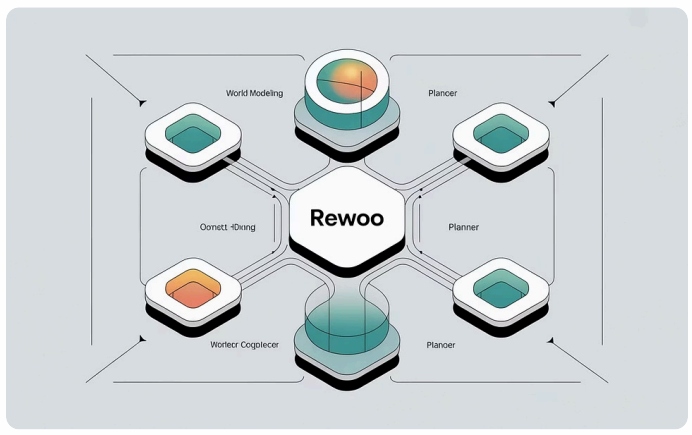
Separates planning from action execution for clearer reasoning.



Reflection Agents

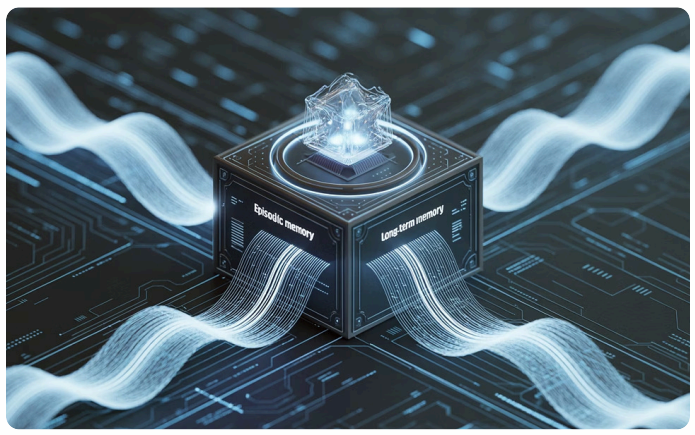
LLMs critique and refine their own outputs for improved reasoning.

Memory systems and reasoning cycles support these architectures:



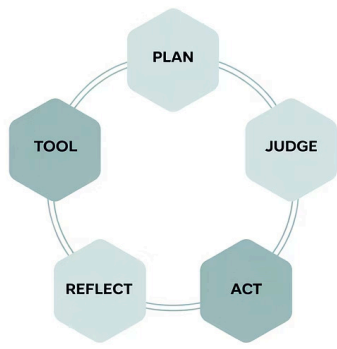
ReWOO

Introduces world modeling + planner modules for enhanced reasoning.



Memory Support

Episodic memory tracks recent steps while long-term memory stores facts or user data.

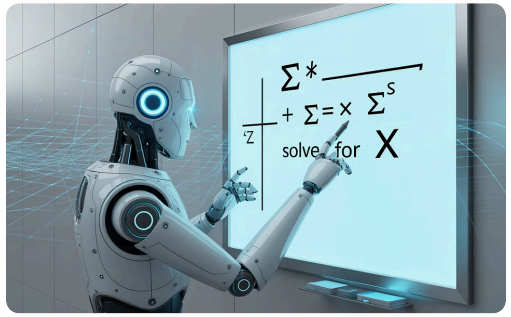


Reasoning Units

The complete reasoning cycle: Plan → Tool → Judge → Reflect → Act.

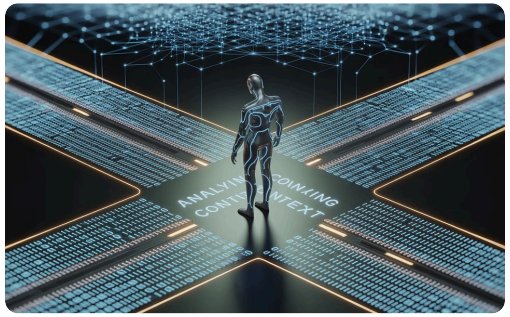
Reasoning Capabilities

What can LLM Reasoners do?



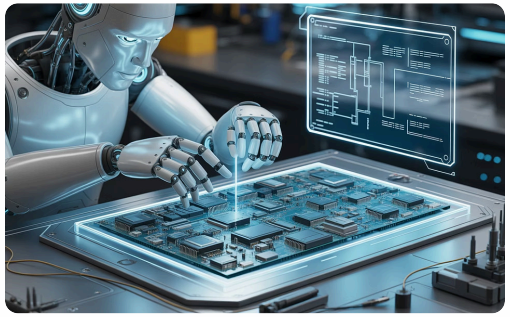
Multistep Problem Solving

Break down math word problems or workflow automation



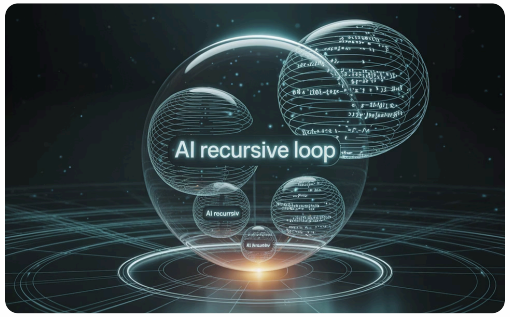
Contextual Decision-Making

Determine whether to search or generate based on context



Self-Correction

Re-analyze incorrect results and adjust approach



Recursive Thinking

Explore possibilities until goal is satisfied

The Battle of the Models

Key Milestones

1

Transformer Architecture

The Transformer architecture revolutionised natural language processing in 2017, allowing models to process information in parallel and handle long-term dependencies more efficiently.

2

BERT Model

The 2018 BERT (Bidirectional Encoder Representations from Transformers) demonstrated the importance of bidirectional pre-training, improving performance on NLP tasks.

3

GPT-3

GPT-3 marked a milestone in 2020 with its size and capability, showing that a large, generalised model could generate coherent, high-quality text in multiple languages and tasks.

4

ChatGPT: Fine-tuning for Instructions

Advances in fine-tuning models for instructions in 2022 improved the ability of LLMs to follow instructions precisely and perform more specialised tasks.

5

Multimodal Models

In 2024, GPT-4 and other multimodal models enabled the integration of multiple types of information.

6

LLM Reasoners

The emergence of LLM reasoners marked a significant advancement in 2025, combining symbolic logic with neural prediction to enable complex planning, self-evaluation, and multi-step problem solving capabilities.

Major Players

OpenAI

NVIDIA®

OpenAI

Pioneers of GPT models and the ChatGPT tool, leading innovation in conversational AI.

Nvidia

Leader in the design and development of advanced graphics processing units (GPUs) that have driven this GenAI revolution

Other Important Actors

Microsoft

Close collaboration with OpenAI. Integration of AI in products like Office and Azure.

Anthropic

Developers of the Claude model. Focus on safe and ethical AI.

Alibaba

Chinese e-commerce giant investing in AI and LLM development.

Meta

Development of "open source" LLMs, including the LLaMA model.

Mistral

French company specialised in the development of open-source models. A reference in the EU.

Google

Leader in AI innovation with search and products

Model Classification

Parameters

Ranging from millions to billions. Influence the capacity and complexity of the model.

Examples: GPT-2 ~ 1.5 billion parameters

GPT-3 ~ 175 billion parameters

Indicated in the name of the model, e.g: Mistral 7B

Context Window

From hundreds to tens of thousands of tokens. Affects the understanding of long texts.

GPT4-turbo: 128,000 tokens (~ 300 pages of text)

Claude 3: 200,000 tokens

Gemini 2.5: 1 million!!

Specialisation

From generalist models to highly specialised ones in specific domains.

Licensing

From proprietary models to (open-source) models

Parameters

1. **SLM (Small Language Model)**

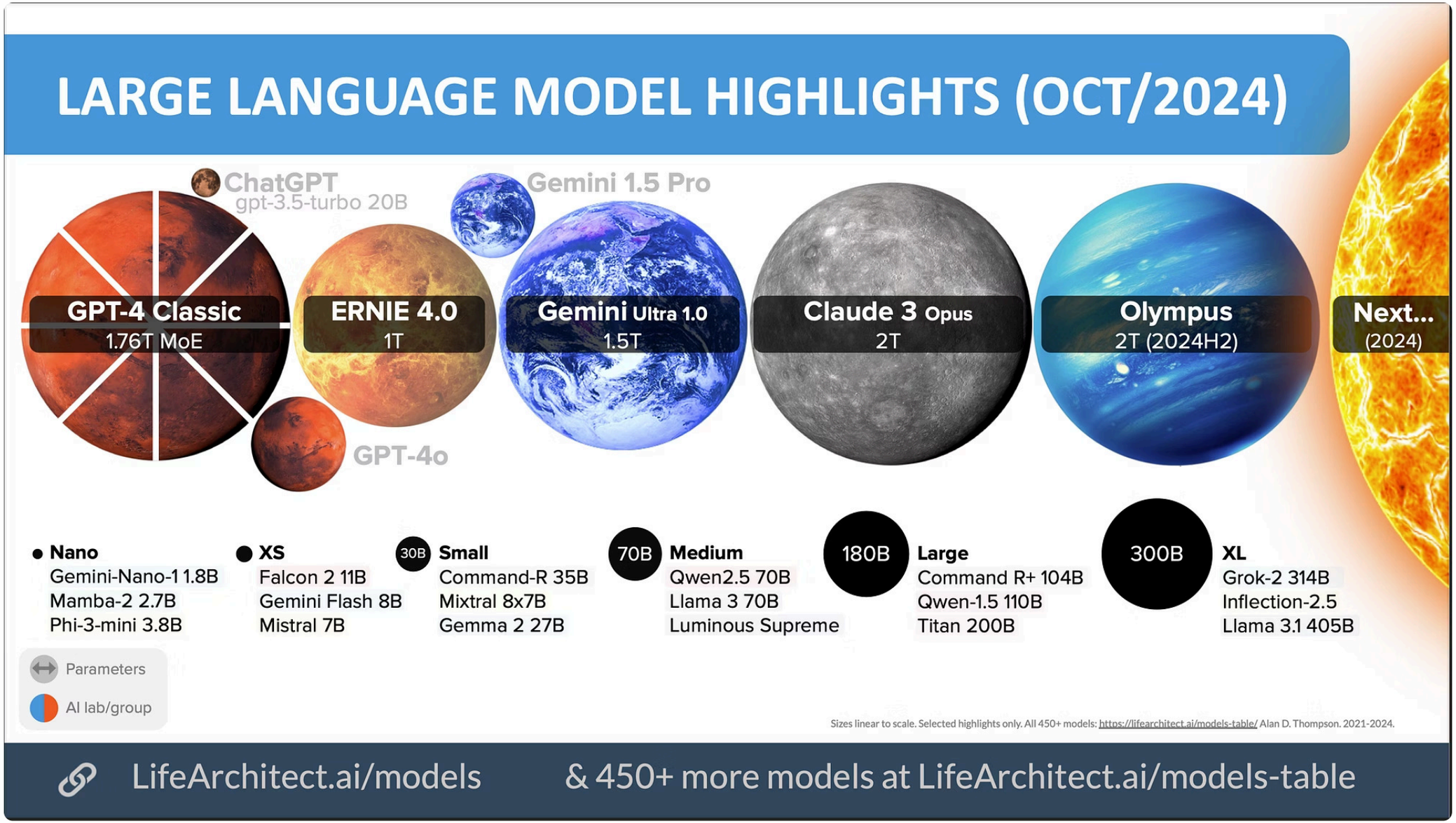
   Approximately 1 billion parameters. Ideal for applications with limited resources.

2. **MLM (Medium Language Model)**

   Around 8 billion parameters. Balance between performance and efficiency.

3. **LLM (Large Language Model)**

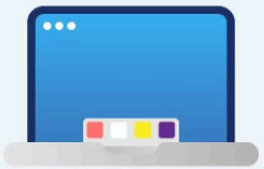
   Around 70 billion parameters. Superior capability for complex tasks.


LARGE LANGUAGE MODEL HIGHLIGHTS (OCT/2024)
ChatGPT
gpt-3.5-turbo 20B
Gemini 1.5 Pro
GPT-4 Classic
1.76T MoE
ERNIE 4.0
1T
Gemini Ultra 1.0
1.5T
Claude 3 Opus
2T
Olympus
2T (2024H2)
Next...
(2024)
GPT-4o
Nano
Gemini-Nano-1 1.8B
Mamba-2 2.7B
Phi-3-mini 3.8B
XS
Falcon 2 11B
Gemini Flash 8B
Mistral 7B
30B
Small
Command-R 35B
Mixtral 8x7B
Gemma 2 27B
70B
Medium
Qwen2.5 70B
Llama 3 70B
Luminous Supreme
180B
Large
Command R+ 104B
Qwen-1.5 110B
Titan 200B
300B
XL
Grok-2 314B
Inflection-2.5
Llama 3.1 405B
Parameters
AI lab/group
Sizes linear to scale. Selected highlights only. All 450+ models: https://lifearchitect.ai/models-table/ Alan D. Thompson. 2021-2024.
LifeArchitect.ai/models
& 450+ more models at LifeArchitect.ai/models-table

FRONTIER AI MODELS + HIGHLIGHTS (MAR/2025)

Microsoft phi
Google Gemma
IBM Granite
Mistral

+ hundreds more...



DeepSeek R
Cohere Command-R
AI21 Jamba
Alibaba Qwen/QwQ

+ hundreds more...

Some images by Flaticon.com. Selected highlights only. All 500+ models: https://lifearchitect.ai/models-table/ Alan D. Thompson. 2021–2025.

LifeArchitect.ai/models-table (500+ model highlights)

Context Window

Amount of input and output information for a model inference.

1. **Short window**: Processes between **512 and 2048 tokens (1 to 5 pages** of text in Spanish). Ideal for brief responses or simple queries.
2. **Medium window**: Handles between **2048 and 8192 tokens (5 to 20 pages)**. Suitable for multi-turn dialogues and moderately long texts.
3. **Long window**: Processes **more than 8K tokens (more than 20 pages)**. Perfect for handling extensive documents or analysing complex information.

Specialisation

Generalist LLMs are capable of performing a wide range of tasks, while specialised ones are designed for a specific purpose.

Hard Prompts

Evaluates the model's ability to handle complex or ambiguous inputs.

Instruction Following

Measures the model's ability to follow and execute precise instructions.

Coding

Evaluates the model's ability to generate and manipulate code.

Math

Measures the model's capacity to solve mathematical problems.

Multi-Turn Conversations

Evaluates the model's ability to maintain coherent and contextualised conversations.

Long Queries

Measures the model's ability to understand and process long inputs.

Logical Reasoning

Evaluates the model's capacity for logical reasoning and drawing valid conclusions.

Memory and Context

Measures the model's ability to remember previous information and maintain context during a conversation.

Proprietary vs. Open Models

Availability:

Proprietary: Developed by companies, with restricted access under API (usually paid) and closed source.

Open: Available for download and installation (on-premises or in the cloud).

Access and Licensing:

Proprietary: Require payment or subscription, with restrictive licenses.

Open: Free and accessible under open licenses, allow customisation.

Control and Customisation:

Proprietary: Limited customisation, controlled by the company.

Open: Full user control to modify and adapt.

Quality and Costs:

Proprietary: High quality, but high costs and restrictions.

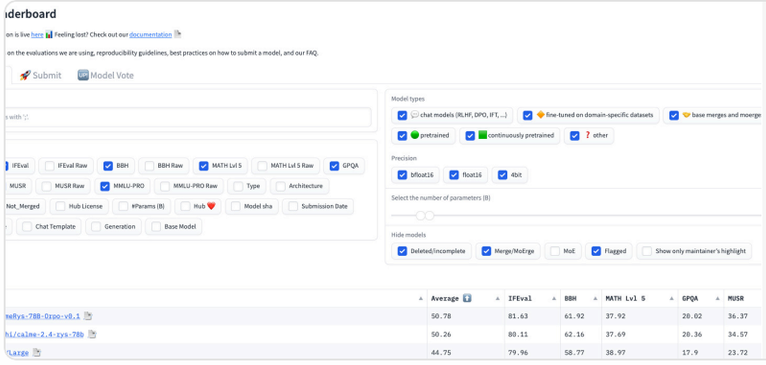
Open: *Less powerful and cheaper to implement.*

Transparency:

Proprietary: Little or no information about the training data

Open: *Offer sufficiently detailed information about the data used to train the system?*, or *The data with which the model has been trained is accessible and available in open format.*

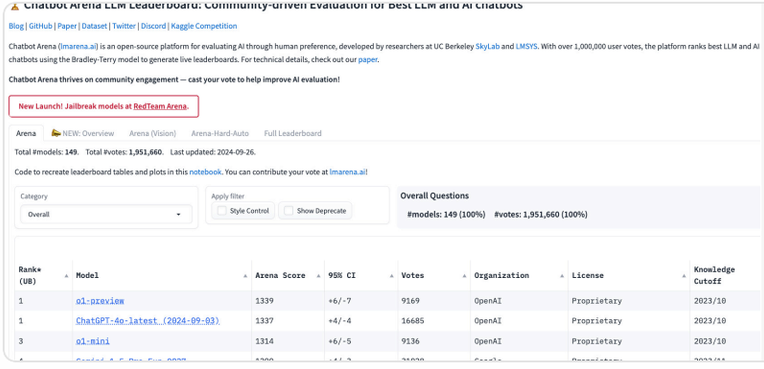
LLM Evaluation



huggingface.co

🤗 Open LLM Leaderboard

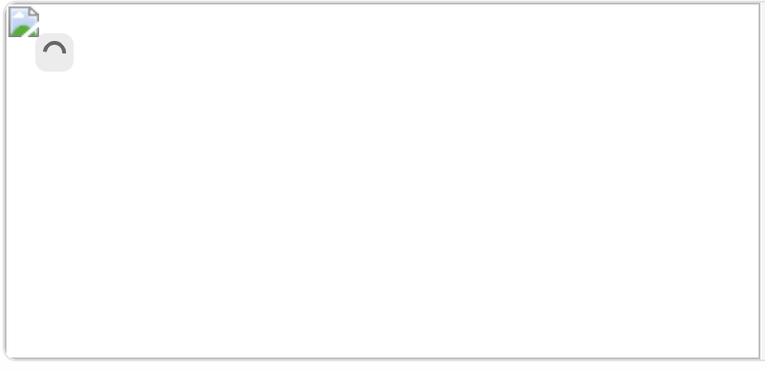
Evaluation of open source Large Language Models (LLMs)



lmarena.ai

Chatbot Arena

Ppen-source platform for evaluating AI through human preference, developed by researchers at UC Berkeley SkyLab and LMSYS. With over 1,000,000 user votes,...



scale.com

SEAL leaderboards

Discover the SEAL LLM Leaderboards for precise and reliable LLM rankings, where leading large language models (LLMs) are evaluated using a rigorous...

Optimization of LLMs

1. Quantization

What it is: Reduces the precision of numbers used in a model — typically from 32-bit floats (`float32`) to 8-bit or 4-bit integers.

Goal: Smaller model size, faster inference, lower memory use.

Example:

- Original weight: `0.124353` (float32)
- Quantized: `0.1` rounding to one-decimal precision

Use Cases:

- Running models on laptops, phones, or edge devices.
- Reducing costs in cloud inference.

Optimization of LLMs

2. Pruning

What it is: Removes parts of the model (e.g., individual weights, neurons, or layers) that contribute little to the final output.

Goal: Reduce model size and computation without large loss in performance.

Types:

- **Unstructured pruning**: Removes individual weights.
- **Structured pruning**: Removes entire neurons or attention heads.

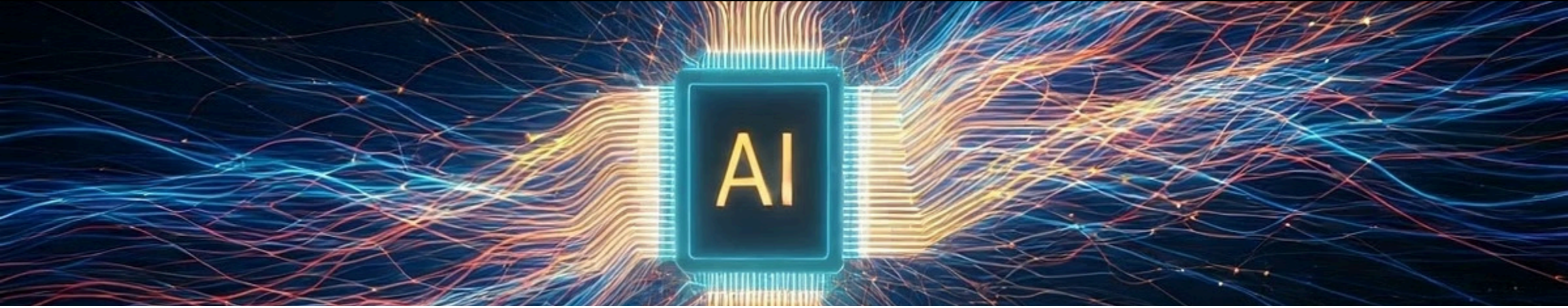
3. Knowledge Distillation

What it is: Trains a smaller model (**student**) to mimic the behavior of a larger one (**teacher**).

Goal: Produce a smaller, faster model that retains much of the performance of a large one.

How it works:

- The student learns not just from the correct answers, but from the teacher model's *soft outputs* (i.e., confidence scores for various answers).



Advantages of Large Models

Improved Coherence

Greater coherence in long responses and better text comprehension.

Ambiguity Detection

Better ability to identify and resolve ambiguities in language.

Relevant Information

Superior ability to provide relevant and contextualised information.

Drawbacks of Large Models

High costs of cloud services

Need for very powerful hardware

High energy consumption

Limited inference speed

Hallucinations

Summary

This presentation provides an introduction to large language models (LLMs), how they work, their advantages and disadvantages. It covers their fundamentals and their use in different fields.

It explores LLM architectures, their training, and the diverse applications they enable, such as machine translation, text generation, and question answering.

Different types of LLMs are analysed, from basic models to conversational and domain-specific models.

These models are based on deep neural networks, which mimic the functioning of the human brain in processing and learning from data. Through techniques like deep learning, language models can capture the nuances and complexities of language, enabling them to generate coherent and contextually relevant text.