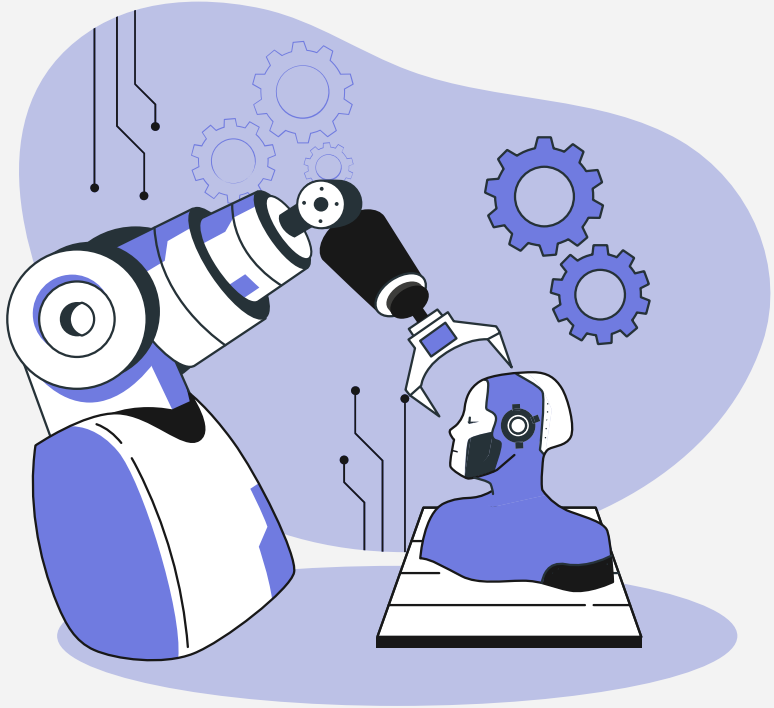
LLM Primer

Nikko Carlo Yabut, MEng AI

Demystifying GenAI:
Large Language Models & Retrieval-Augmented Generation

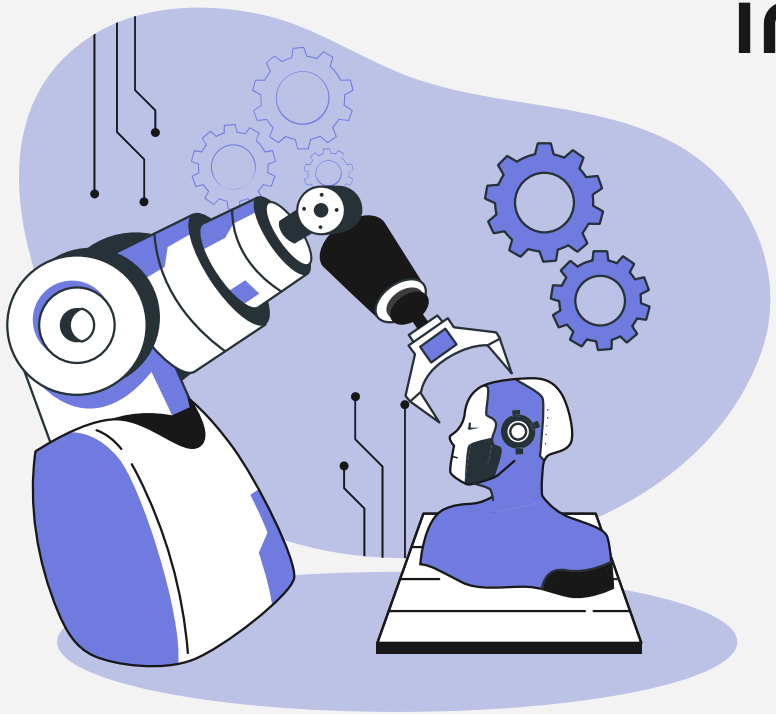
Nikko Carlo Yabut, MEng AI
Certified: AWS Certified Machine Learning - Specialty

Contents

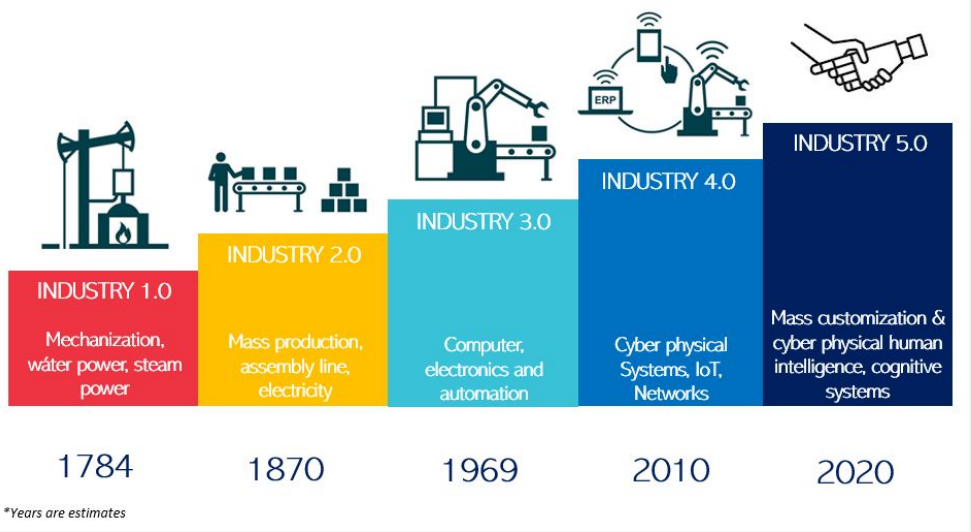


01 Intro

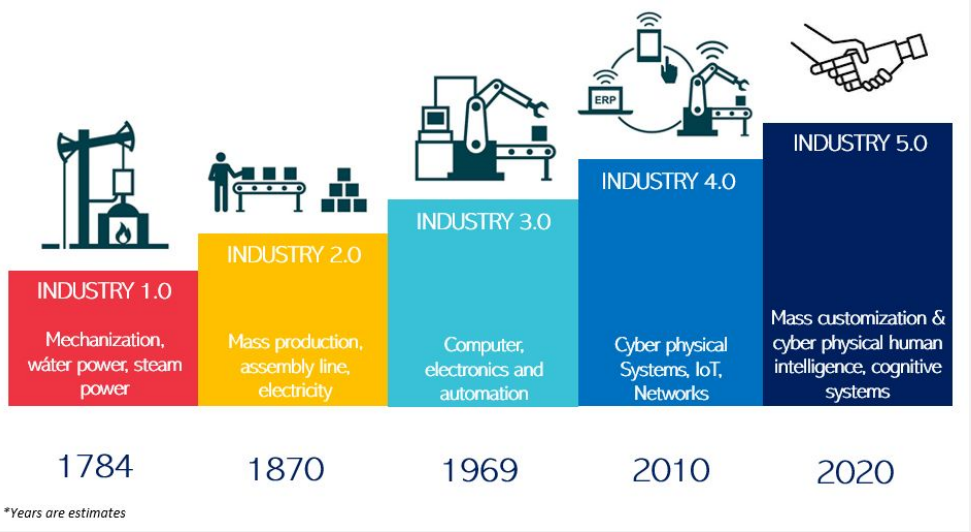
Where are we now in the Industrial Revolution?



Evolution of Industrial Revolution



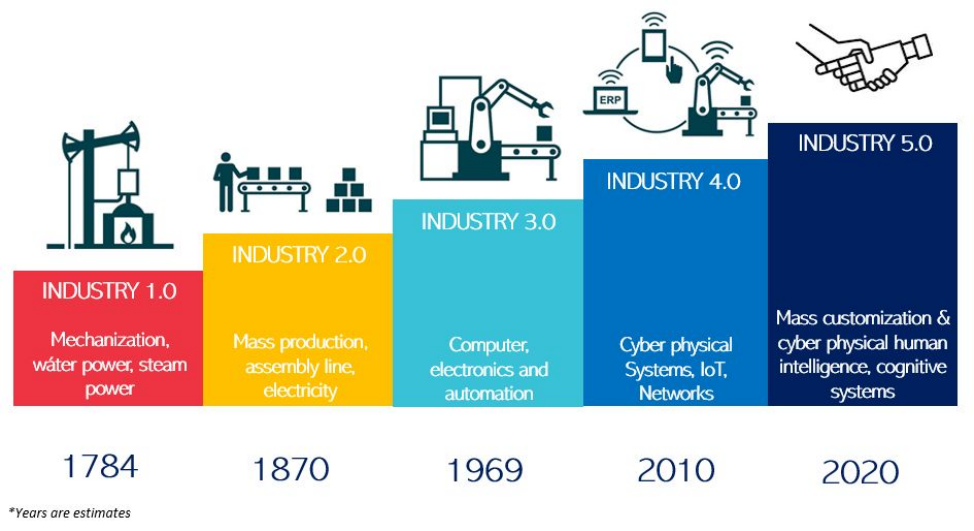
Evolution of Industrial Revolution

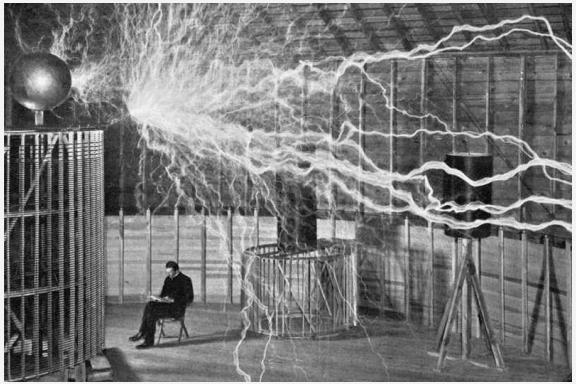


Industry 1.0: The First Industrial Revolution

- **From MANUAL labor to Machines**
- **Key Technology:** Water and steam power.
- **Impact:**
  - Machines transformed manufacturing, starting with textiles in Europe.
  - Fueled by coal, industrial growth expanded significantly.
  - The economy saw a massive boost, and large-scale production became possible.
- **Outcome:** Marked the beginning of large-scale industrialization.

Evolution of Industrial Revolution

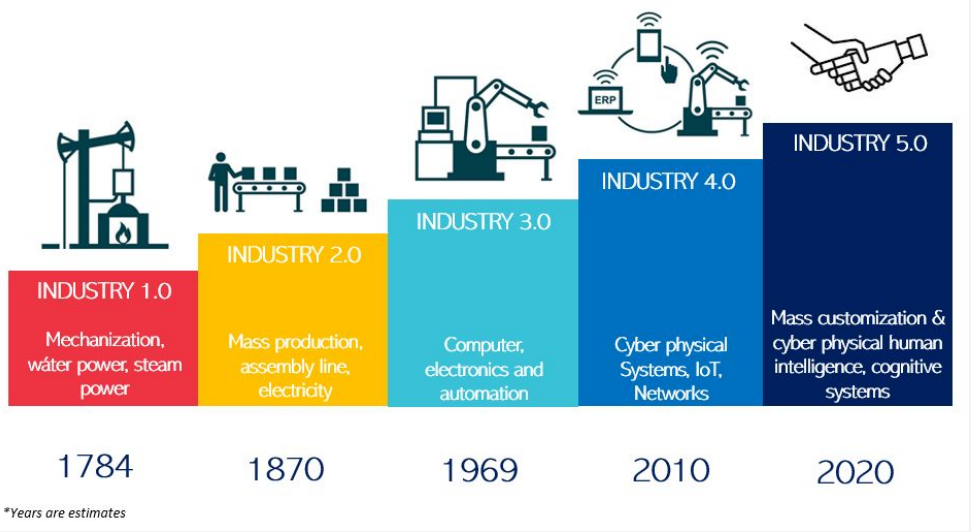




Industry 2.0: The Second Industrial Revolution

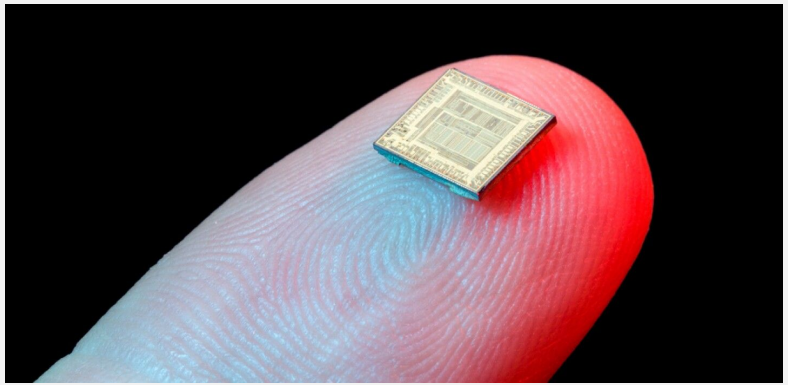
- **Key Technology:** Electrification.
- **Impact:**
  - **Electricity replaced steam** as the primary energy source for machines.
  - Mass distribution of **electricity enabled advanced manufacturing**.
  - Technological advancements included assembly lines, and publications like *The Principles of Scientific Management* emerged.
  - **Led to increased productivity**, meeting the needs of customers, employees, shareholders.
- **Outcome:** Often called the **Technology Revolution** due to the scale of innovation.

Evolution of Industrial Revolution

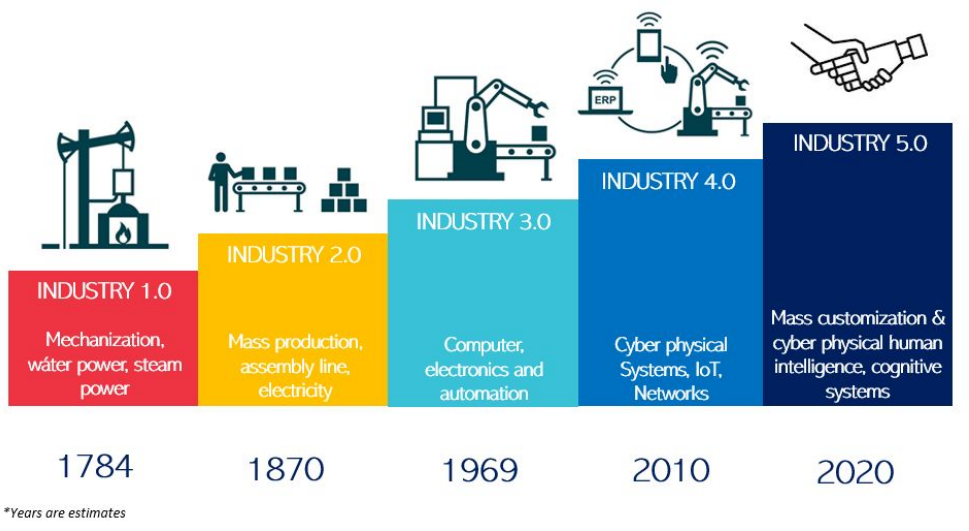


Industry 3.0: The Third Industrial Revolution

- **Key Technology:** Microchips, computers, and automation.
- **Impact:**
  - Introduction of computers facilitated **automation in production.**
  - **Programmable Logic Controllers (PLC)** replaced human workers in assembly lines.
- **Outcome:** Known as the **Information Revolution** or **IT Revolution**, due to the rise of computing and digital technologies.



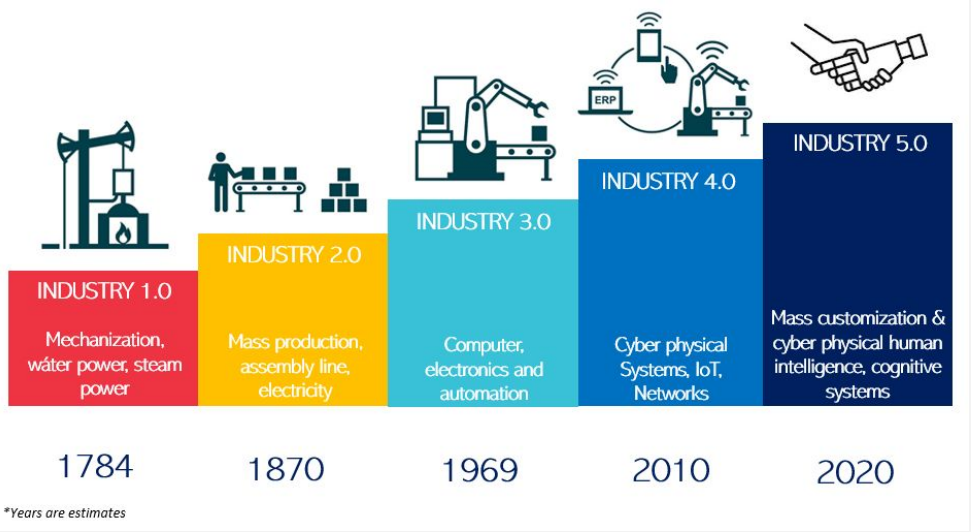
Evolution of Industrial Revolution



Industry 4.0: The Fourth Industrial Revolution

- **Key Technologies:**
  - **Telecommunications (Wireless Comms)**
  - **Internet**
  - **Internet of Things (IoT)**
  - **Cloud Computing**
  - **Big Data**
  - **Robotics**
  - **AI**
- **Impact:**
  - Fully automated production systems where machines communicate, control, and make decisions **without human intervention.**
- **Outcome:** Known as the **Digital Revolution**, reshaping industries and leading to higher levels of efficiency.

Evolution of Industrial Revolution



Industry 5.0: The Fifth Industrial Revolution

- **Time Period:** Emerging (futuristic vision).
- **Key Focus:** Collaboration between human and AI.
- **Impact:**
  - Emphasizes **co-bots** (collaborative robots) working alongside humans.
  - Enhanced efficiency through **human-robot collaboration on the factory floor.**
  - Inspired by Japan's **Society 5.0** concept, where humans and AI deliver business value together.
- **Outcome:** A more human-centric approach.

Evolution of Industrial Revolution

Where are we now in the Industrial Revolution?

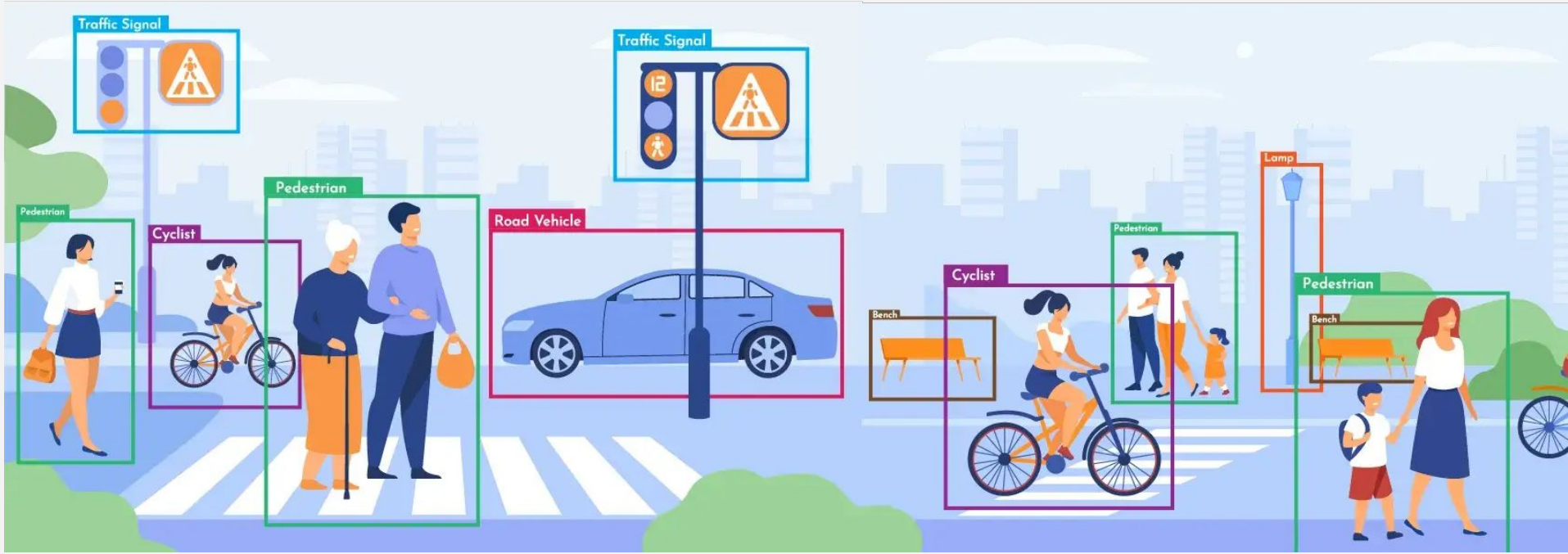
Industry 4.0
(Digital Revolution)

&

Industry 5.0
(Human-AI Collaboration)

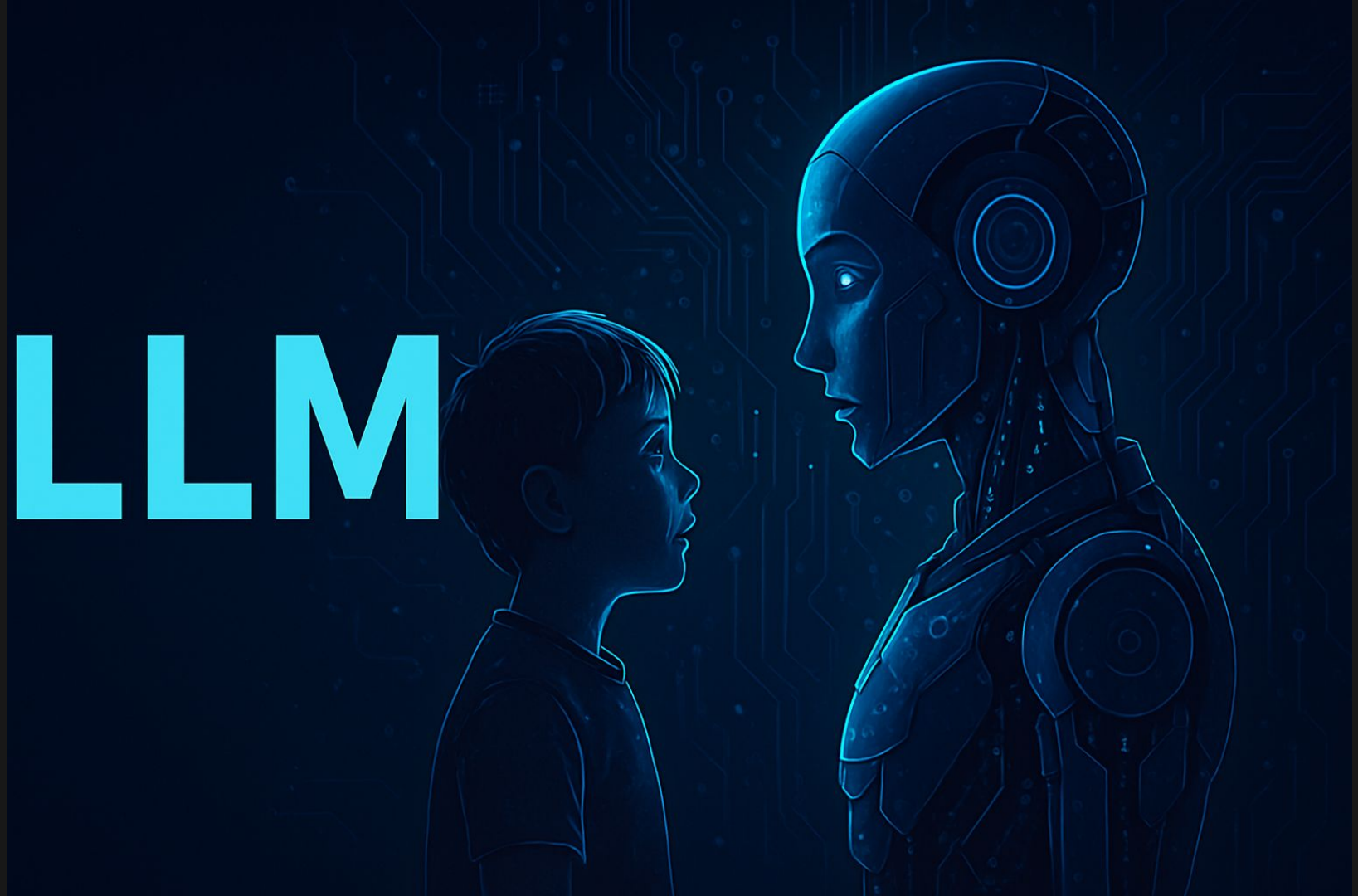
Evolution of Industrial Revolution

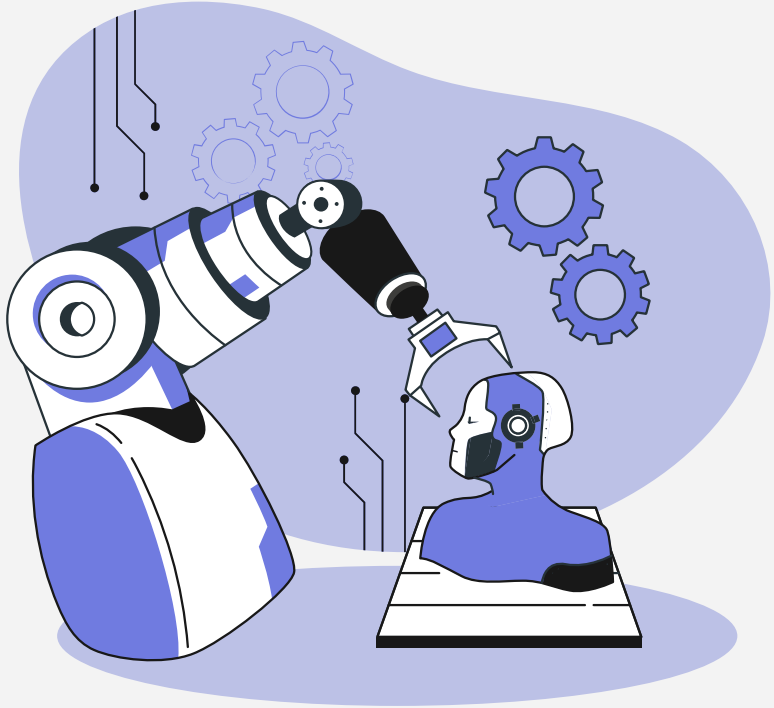
In this discussion, we will focus on a specific use case of Industry 5.0: LLMs



NLP

- Part-of-Speech (POS) Tagging
- Emotion & Intent Detection
- Named Entity Recognition (NER)
- Text Classification
- Information Extraction
- Text Similarity
- LLM


LLM



02
AI, ML, DL, GenAI

Intelligence

A very general mental capability that among other things involves the ability to:

- Reason
- Plan
- Solve problems
- Think abstractly
- Comprehend complex ideas
- Learn from experience

Journal of Intelligence 1997 Vol 24 No 1

Intelligence

A very general mental capability that among other things involves the ability to:

- Reason
- Plan
- Solve problems
- Think abstractly
- Comprehend complex ideas
- Learn from experience

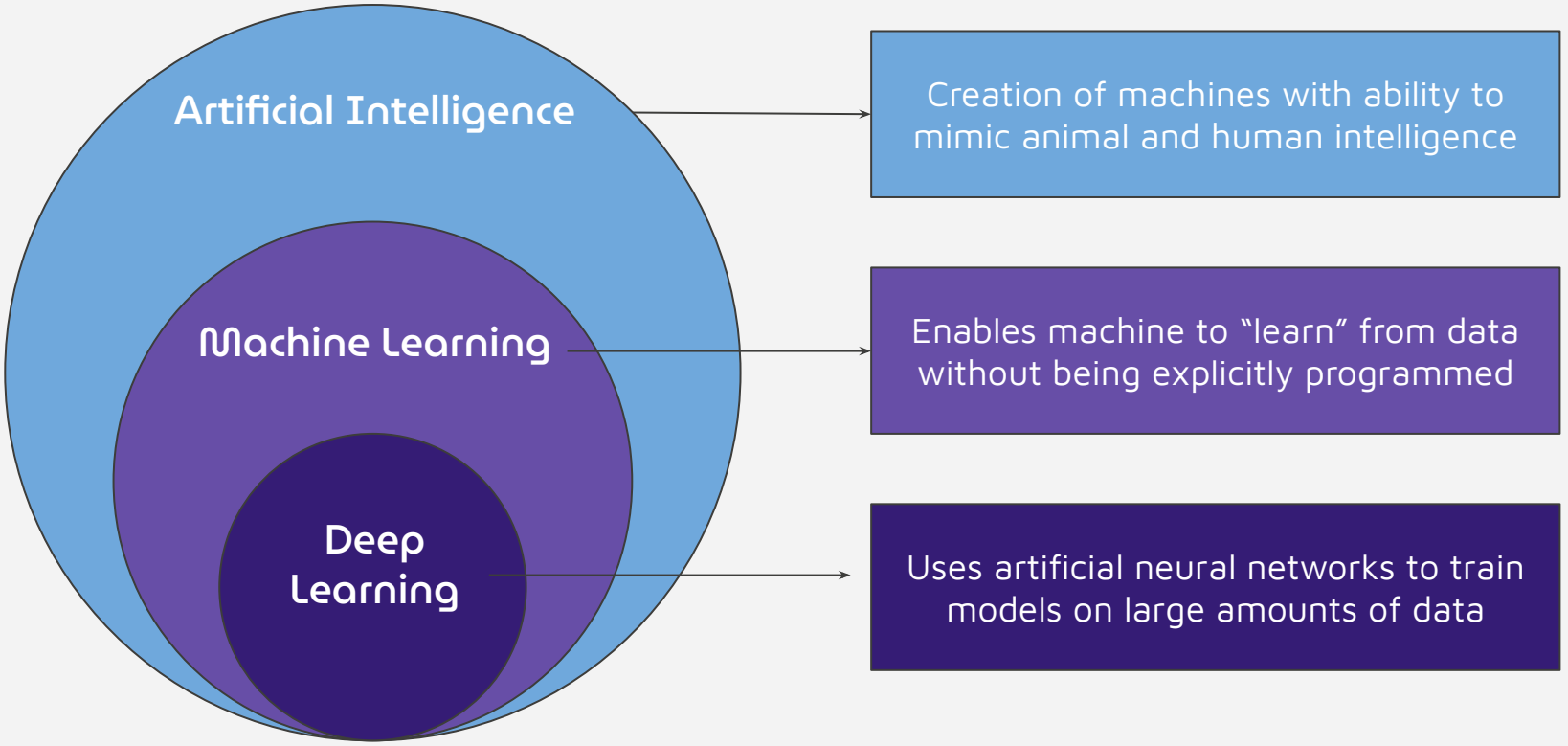
Journal of Intelligence 1997 Vol 24 No 1

Artificial Intelligence (AI)

A machine has AI capabilities
IF it exhibits **animal** or **human** intelligence

Ants bridging algo

AI, ML, DL


Artificial Intelligence
Machine Learning
Deep Learning
Creation of machines with ability to mimic animal and human intelligence
Enables machine to "learn" from data without being explicitly programmed
Uses artificial neural networks to train models on large amounts of data

AI, ML, DL, GenAI

Artificial Intelligence

Is the field of study that deals with creation of machine that exhibits animal or human intelligence.

Machine Learning

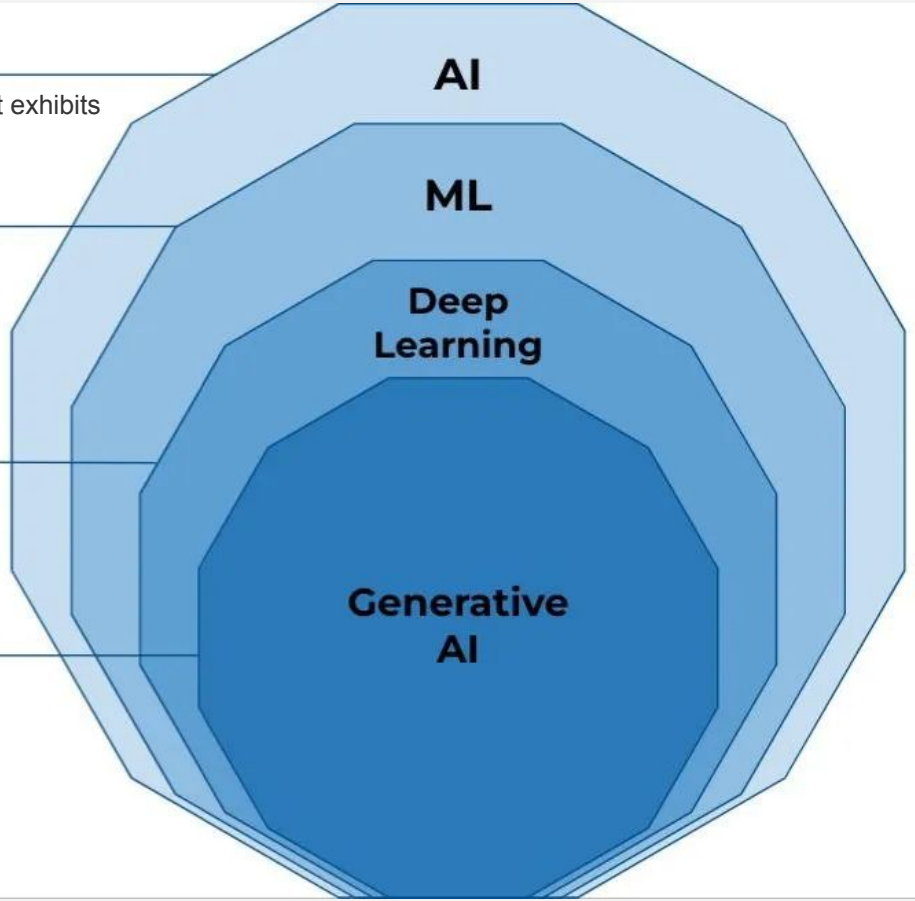
Is a branch of AI that focus on the creation of intelligent machines that learn from data. Another very well know branch inside AI is **Optimization**.

Deep Learning

Is a subset of Machine Learning methods, based on **Artificial Neural Networks.**
Examples: CNNs, RNNs

Generative AI

A type of ANNs that generate data that is similar to the data it was trained on.
Examples: GANs, LLMs



AI, ML, DL, GenAI

Artificial Intelligence

Is the field of study that deals with creation of machine that exhibits animal or human intelligence.

Machine Learning

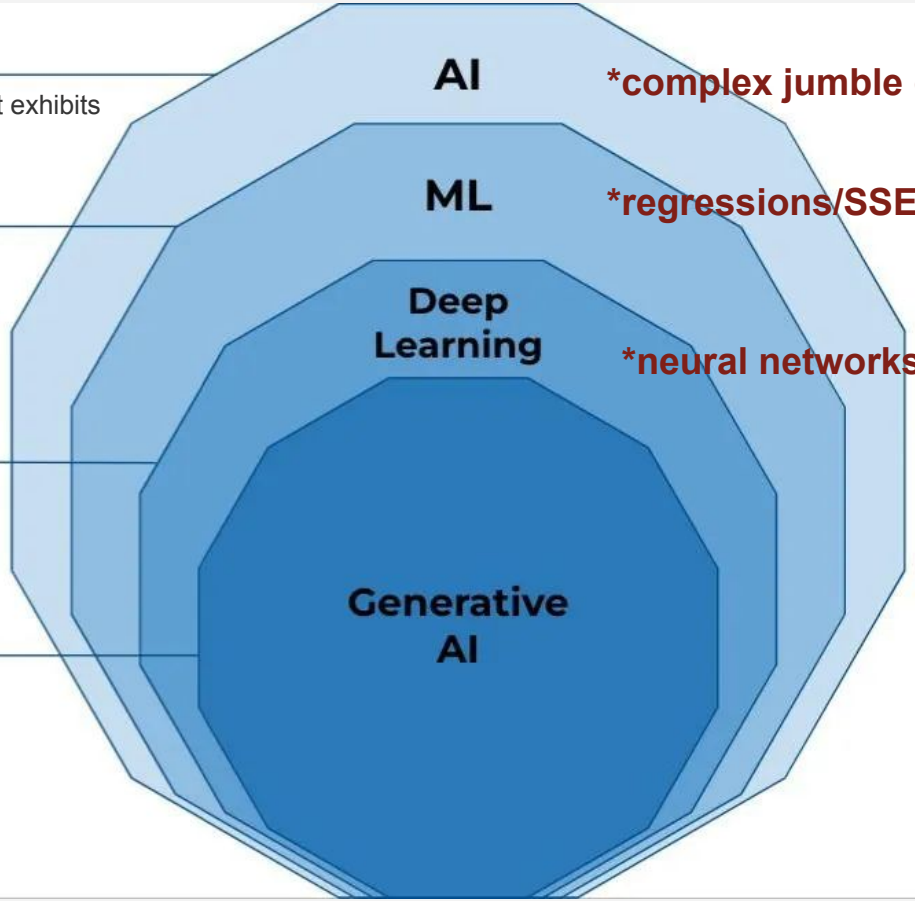
Is a branch of AI that focus on the creation of intelligent machines that learn from data. Another very well know branch inside AI is **Optimization**.

Deep Learning

Is a subset of Machine Learning methods, based on **Artificial Neural Networks.**
Examples: CNNs, RNNs

Generative AI

A type of ANNs that generate data that is similar to the data it was trained on.
Examples: GANs, LLMs



*complex jumble of if/else

*regressions/SSE (linear/log)

*neural networks

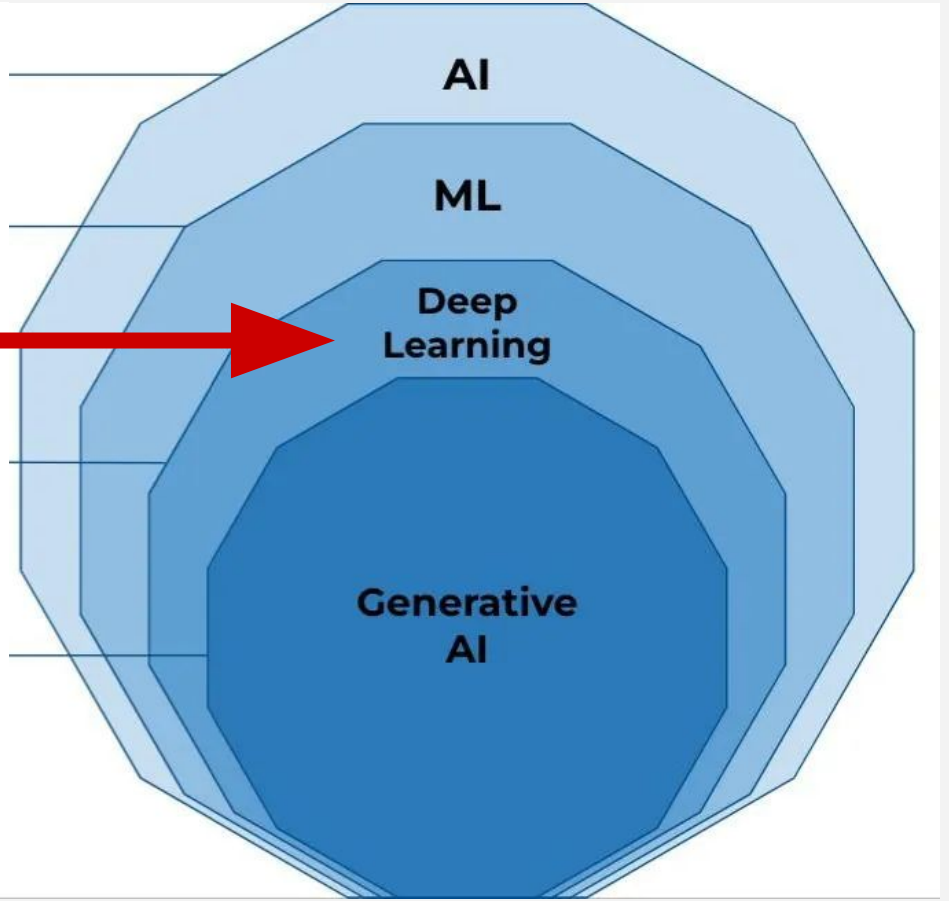
AI, ML, DL, GenAI

The Nobel Prize in Physics 2024

Ill. Niklas Elmehed © Nobel Prize Outreach
John J. Hopfield
Prize share: 1/2

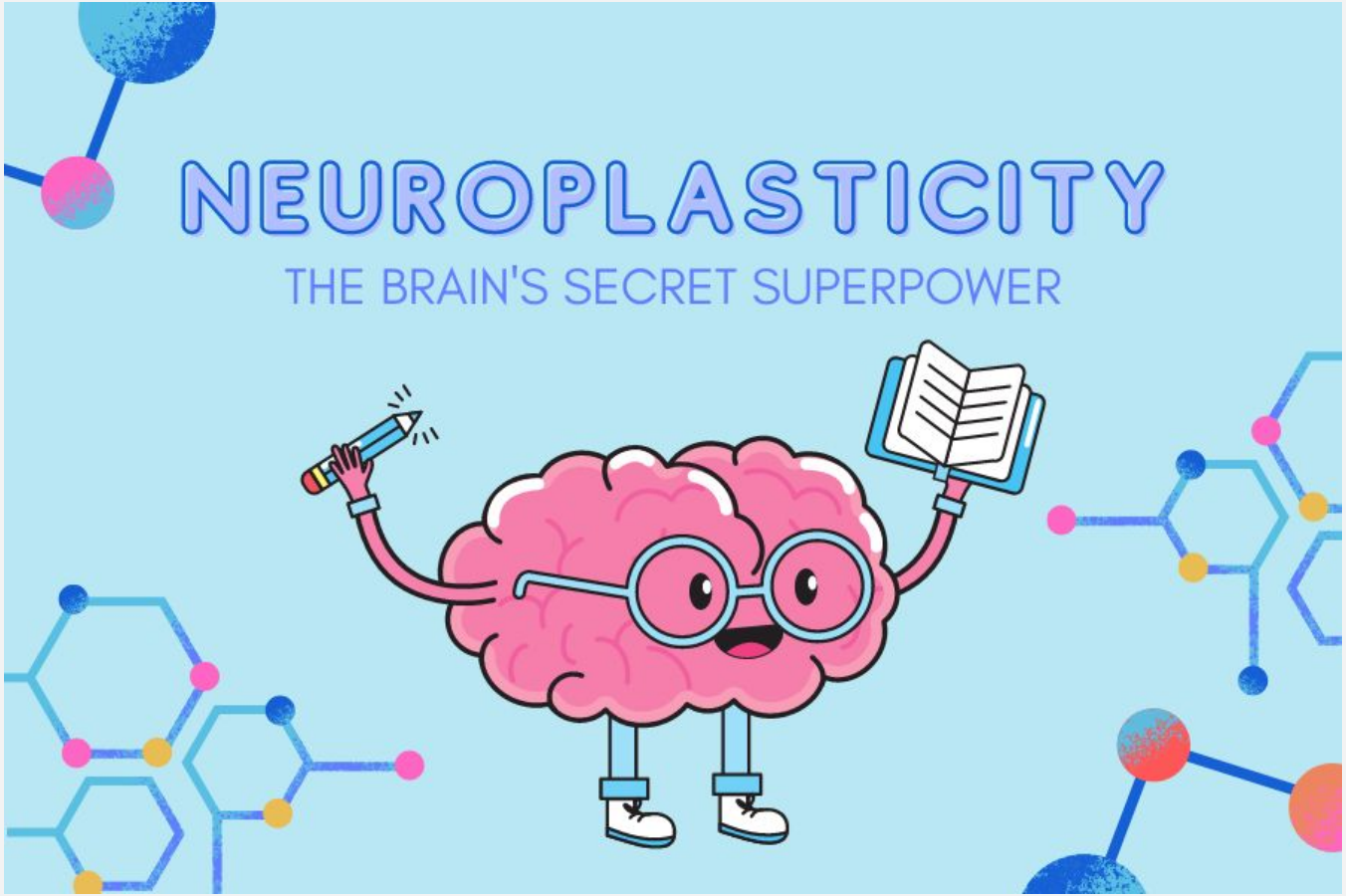
Ill. Niklas Elmehed © Nobel Prize Outreach
Geoffrey Hinton
Prize share: 1/2

The Nobel Prize in Physics 2024 was awarded jointly to John J. Hopfield and Geoffrey E. Hinton "for foundational discoveries and inventions that enable machine learning with artificial neural networks"

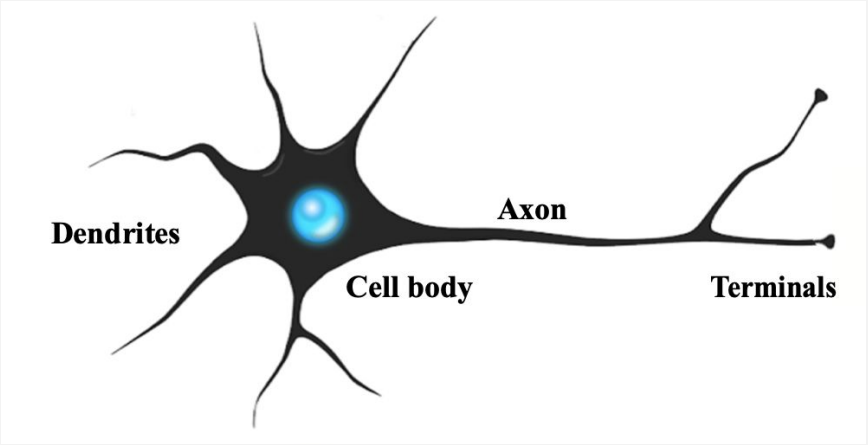


How do machines ‘learn’?

From biological neuron to Neural Network neuron



From biological neuron to Neural Network neuron



Biological Neurons: Structure and Function

Dendrites

Receive signals from other neurons, starting the communication process.

Soma (Cell Body)

Processes incoming signals to decide the neuron's response.

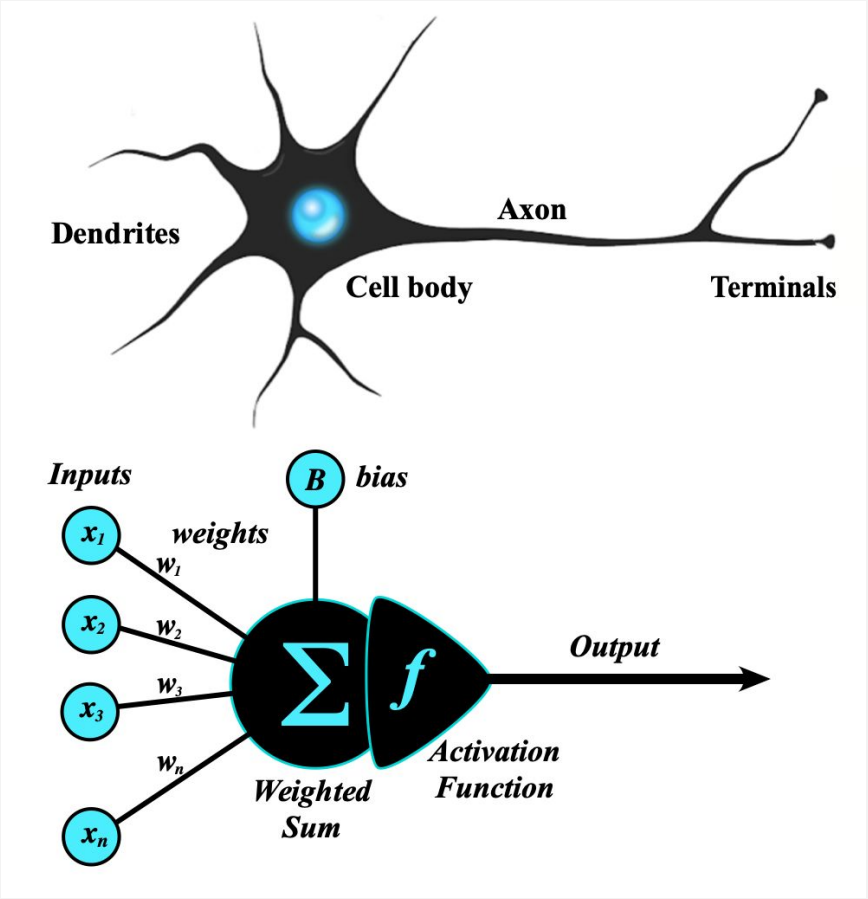
Axon

Transmits the processed signal outward to connected neurons.

Synapses

Junction points where signals pass to neighboring neurons.

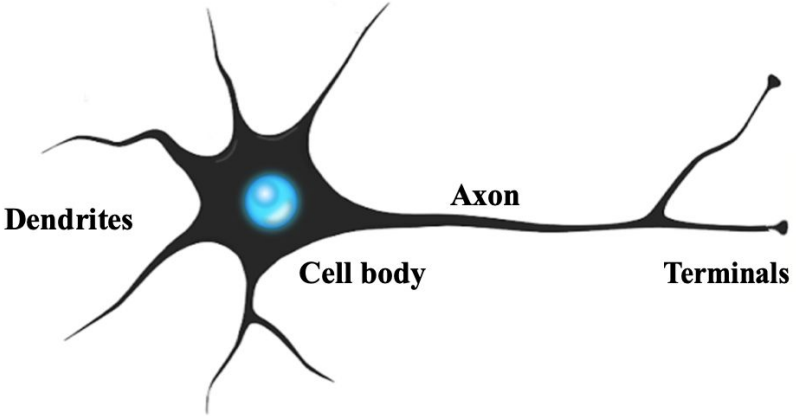
From biological neuron to Neural Network neuron

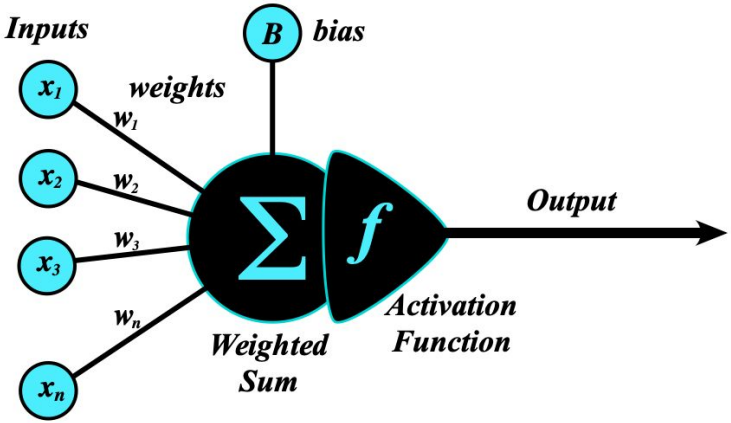


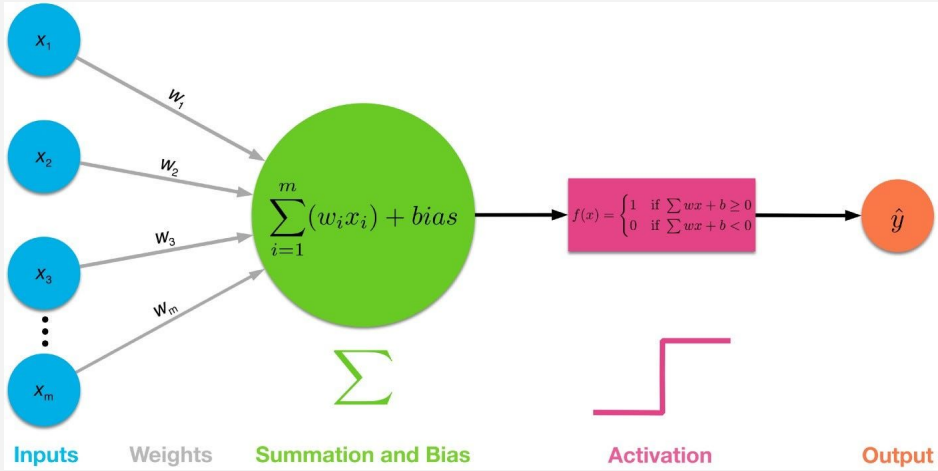
Comparison of Biological Neurons and AI Neurons

- **Structure:**
  - **Biological Neuron:**
    - Composed of dendrites, a cell body (soma), and an axon.
    - Complex connections and synapses with other neurons.
  - **AI Neuron:**
    - Simplified mathematical model or node in a neural network.
    - Represents inputs, weights, activation functions, and outputs.
- **Learning Mechanism:**
  - **Biological Neuron:**
    - Learns through neuroplasticity / synaptic plasticity (strengthening/weakening of connections).
    - Involves complex biological processes like long-term potentiation.
  - **AI Neuron:**
    - Learns through adjustments of weights via algorithms (e.g., backpropagation).
    - Utilizes large datasets to optimize performance during training.

From biological neuron to Neural Network neuron







Linear regression is essentially a special case of a **one-neuron network.**

Linear Regression

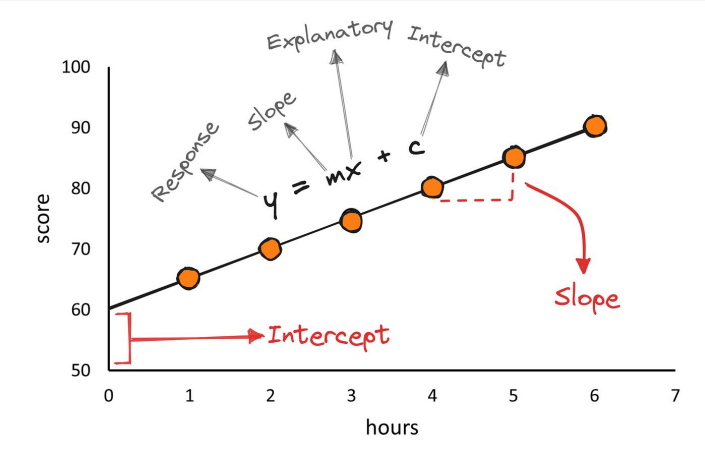
Examples:

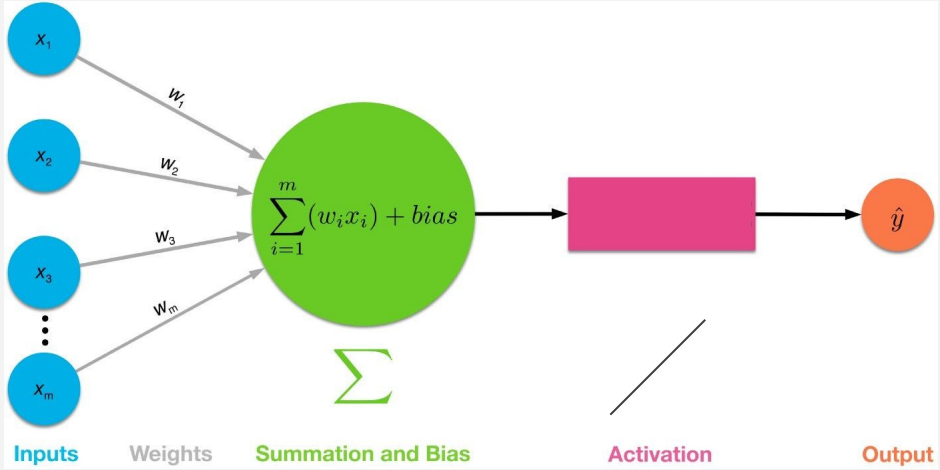
🏠 **1. Housing Price Prediction**

- **X: Features:** Square footage, number of bedrooms, location, age of the house.
- **y: Target:** House price in dollars.

📈 **2. Sales Forecasting**

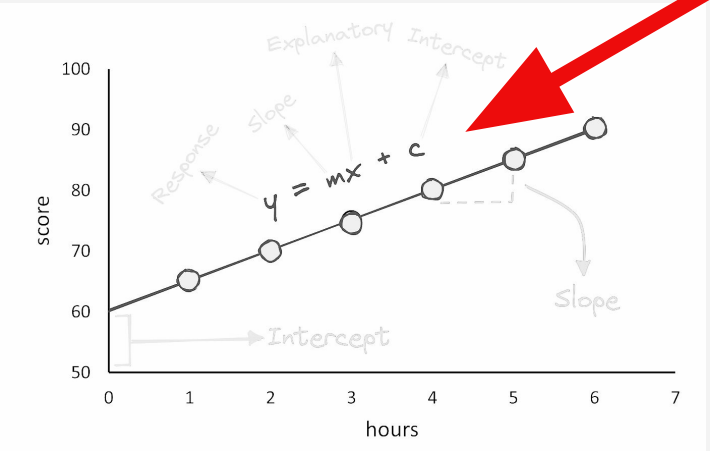
- **X: Features:** Advertising spend, season (month), store location.
- **y: Target:** Sales in units or revenue.

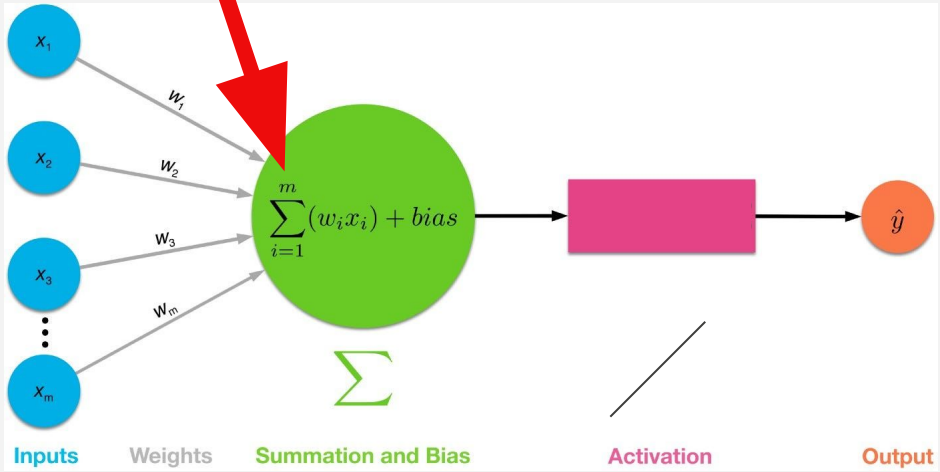




Linear regression is essentially a special case of a **one-neuron network.**

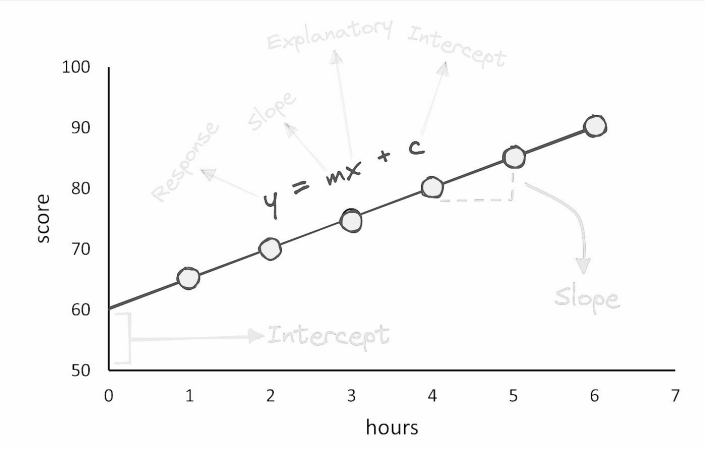
Similar Mathematical Formula!!!

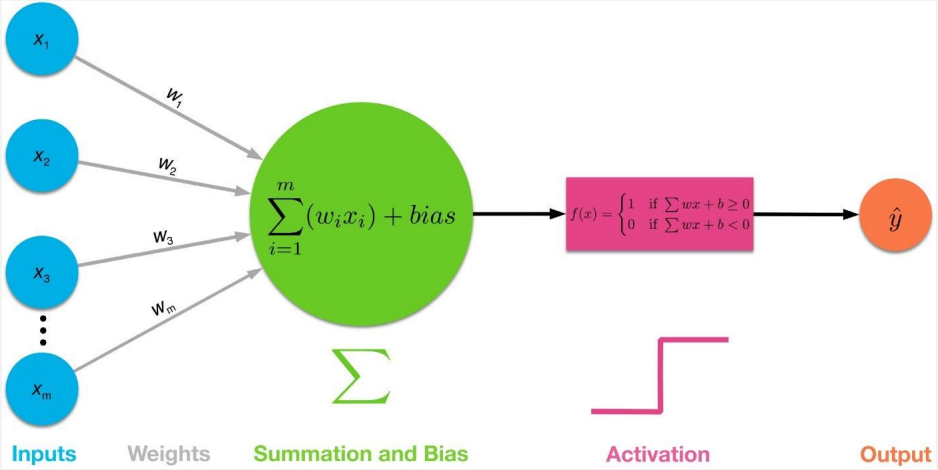




However, the real world is nonlinear!

To truly capture the complexity of the real world, we need networks of to go beyond linearity.





Linear regression is essentially a special case of a **one-neuron network.**

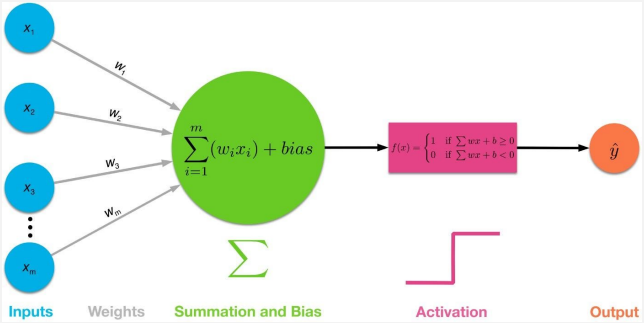
Why One Neuron Isn't Enough

Single neuron limits

Can only solve simple, linear problems.

Complex patterns need more

One neuron can't recognize intricate shapes or boundaries.



Aspect	Linear Regression	One-Neuron Network
Formula	$\hat{y} = \mathbf{w}^\top \mathbf{x} + b$	$y = \phi(\mathbf{w}^\top \mathbf{x} + b)$
Activation Function	None (or identity)	Typically nonlinear (e.g. ReLU, sigmoid)
Training Loss	MSE	Depends (MSE, cross-entropy, etc.)
Use Case	Regression	Regression or classification

✅ 1. Formula for Linear Regression

The **linear regression** model predicts a continuous outcome based on one or more input features:

$$\hat{y} = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b$$

Or more compactly using vector notation:

$$\hat{y} = \mathbf{w}^\top \mathbf{x} + b$$

Where:

- $\hat{y}$ is the predicted output,
- $\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top$ is the input vector,
- $\mathbf{w} = [w_1, w_2, \ldots, w_n]^\top$ is the weight vector,
- b is the bias term (intercept).

✅ 2. Formula for a One-Neuron Network (Perceptron without activation)

A **single neuron** in a neural network (without activation) also computes a weighted sum of its inputs:

$$y = \mathbf{w}^\top \mathbf{x} + b$$

If it includes a **nonlinear activation function** ϕ, the formula becomes:

$$y = \phi(\mathbf{w}^\top \mathbf{x} + b)$$

🔗 3. Relationship Between the Two

Linear regression is essentially a **special case** of a one-neuron network:

- It uses **no activation function** (or a linear activation: $\phi(x) = x$).
- It's trained using **least squares** loss.
- The goal is to minimize the **mean squared error** (MSE) between predictions and actual values.

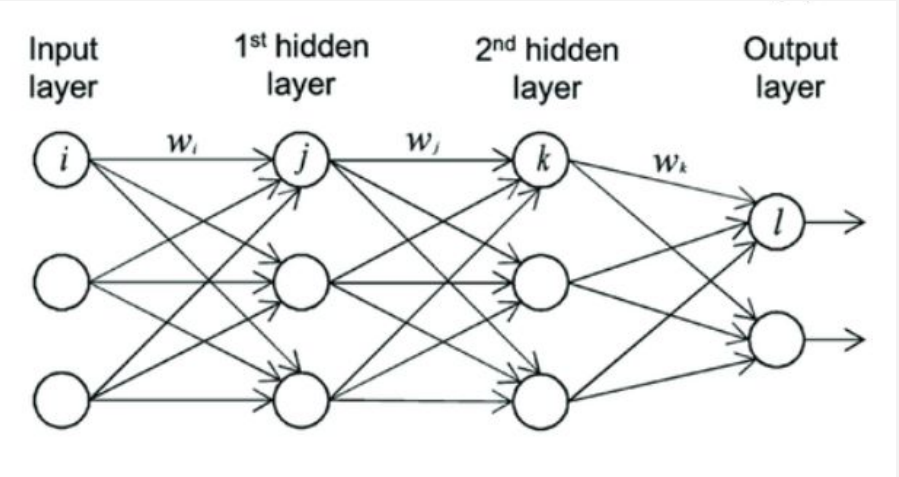
Introducing MLPs: From one neuron to network of neurons

MLP (Multi-Layer Perceptron)?

A **Multi-Layer Perceptron (MLP)** is a type of neural network that **combines many neurons** arranged in **layers** to solve more complex problems than a single neuron can handle.

Just like people working together can solve bigger problems than one person alone, **MLPs stack neurons** into:

- **Input Layer** – receives the data
- **Hidden Layers** – extract patterns and features
- **Output Layer** – produces the final decision or prediction



Each neuron in a layer is connected to **all** neurons in the next layer — this is called a **fully connected network**.

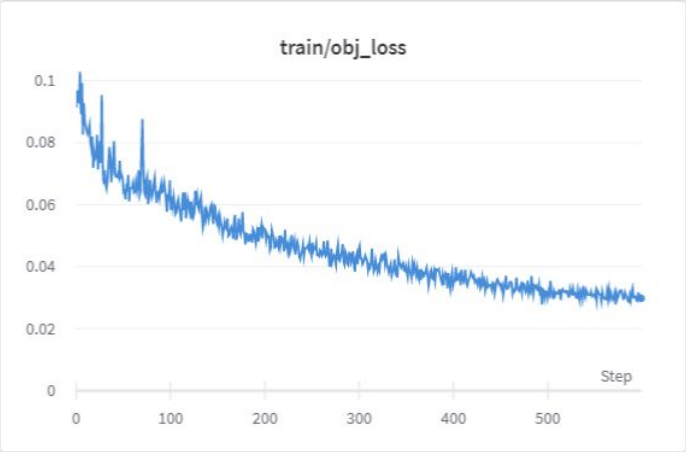
An MLP allows us to go from **simple decisions (single neuron)** to **complex pattern recognition**, by letting multiple neurons **collaborate** and **refine** their outputs through layers.

We’ve built the brain.
Now the big question is…

can it learn?

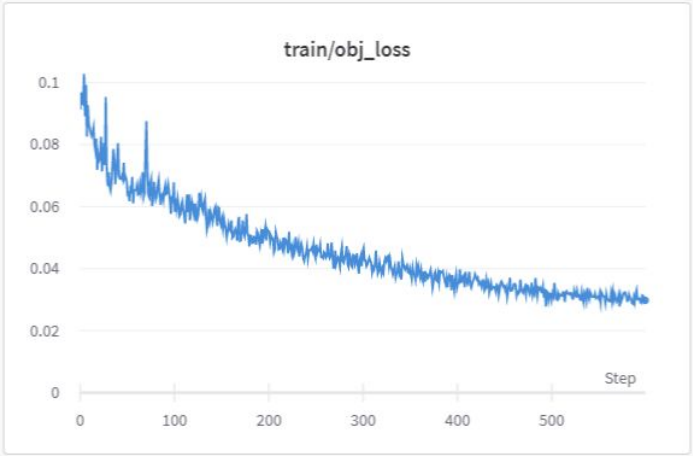
How do machines ‘learn’?

“Learning from mistakes…”



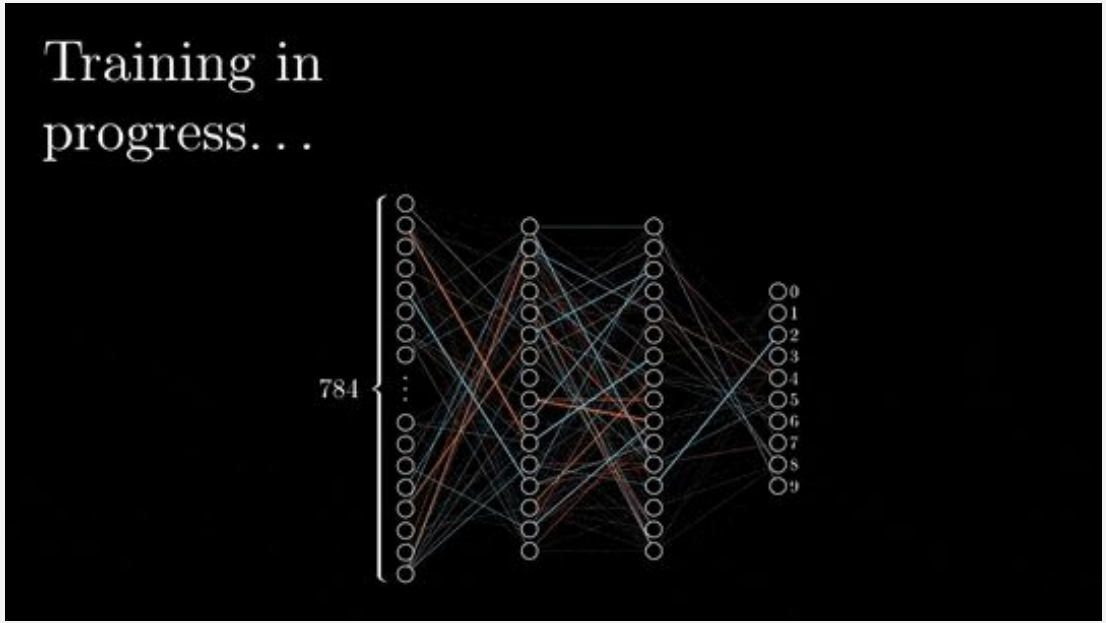
How do machines ‘learn’?

“Learning from mistakes…”



A computer program is said to **learn** from **experience E** with respect to some class of **tasks T**, and **performance measure P**, if its performance at tasks in T, as measured by P, improves with experience E. – Tom Mitchell

How does machines ‘learn’? BACKPROPAGATION



A computer program is said to **learn** from **experience E** with respect to some class of **tasks T**, and **performance measure P**, if its performance at tasks in T, as measured by P, improves with experience E. – Tom Mitchell

What is Backpropagation?

Backpropagation is how a neural network **learns from its mistakes**.

It works by:

1. **Comparing** the model’s prediction to the correct answer (using a **loss function**).
2. **Measuring the error**, then

Sending that error backward through the network to **adjust the weights**—so it does better next time.

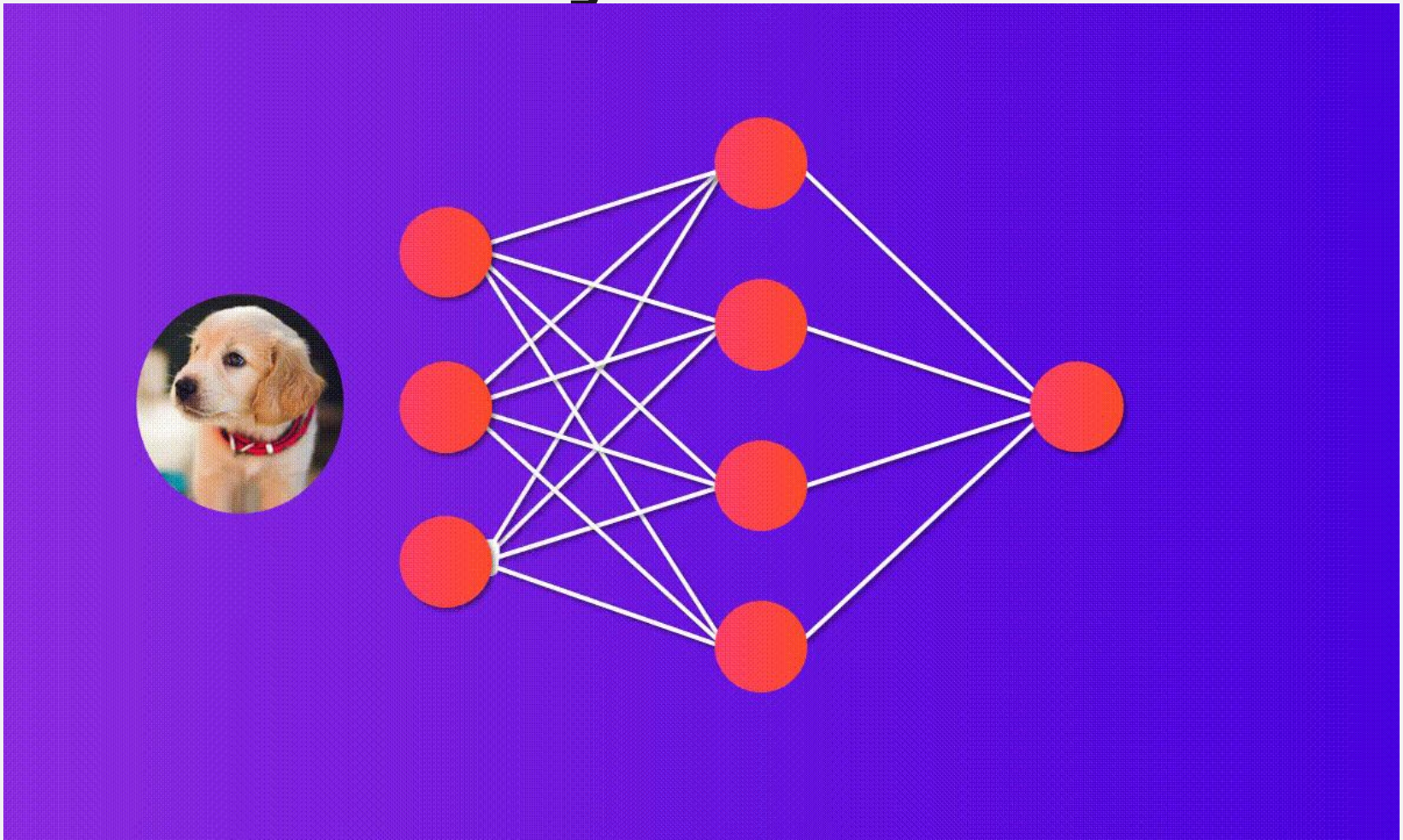
💡 **Analogy:**
Like a student reviewing a wrong answer, figuring out *why* it was wrong, and correcting their approach.

How does machines ‘learn’?

“Learning from mistakes…”

A computer program is said to **learn** from **experience E** with respect to some class of **tasks T**, and **performance measure P**, if its performance at tasks in T, as measured by P, improves with experience E. – Tom Mitchell

Neural Network doing Prediction



Types of Learning Problems

Supervised Learning

These are images of dogs.

These are images of cars.

Now, what is this an image of?

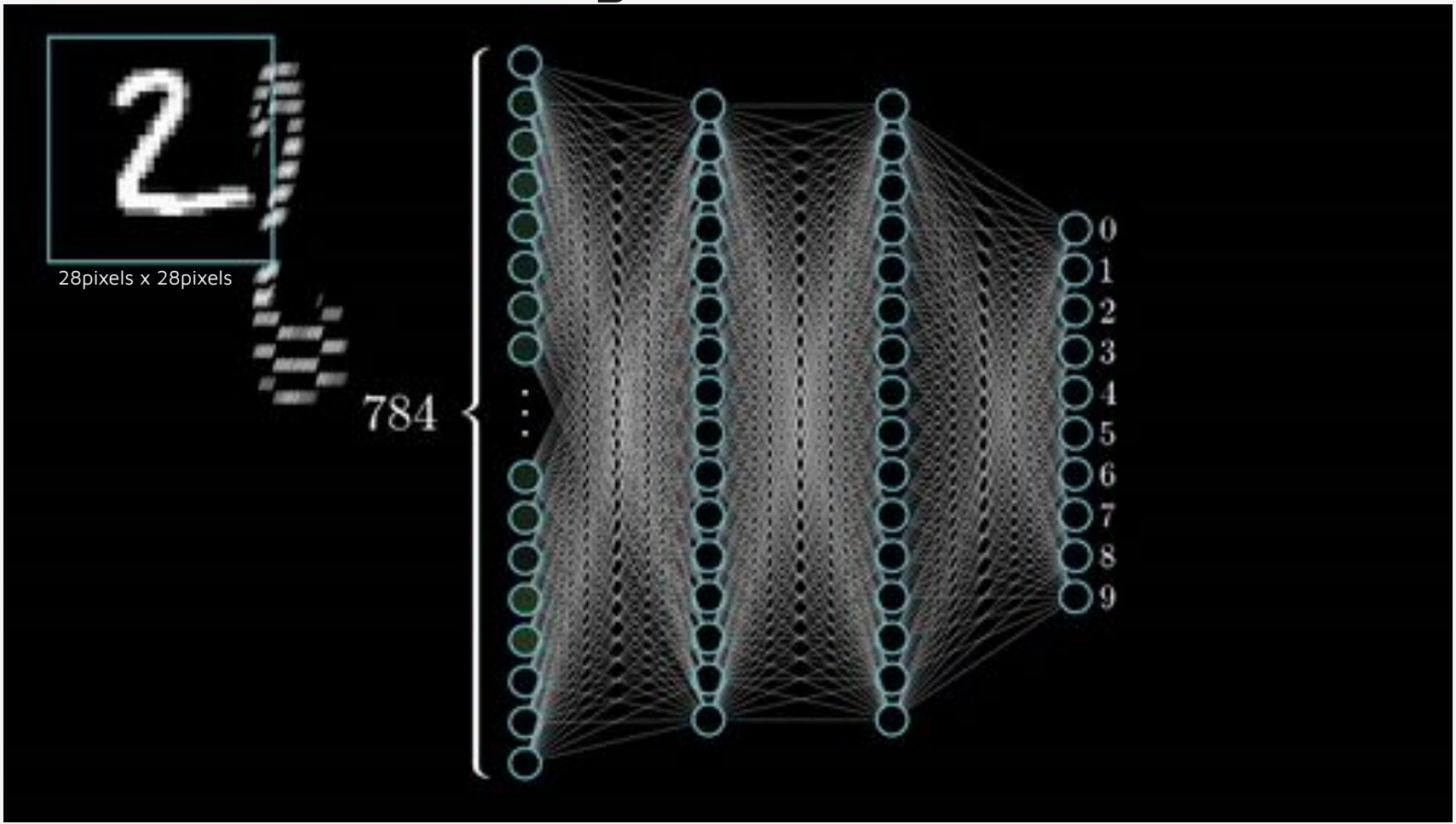
Unsupervised Learning

Here are some images...

Is there an image that does not belong?

Are there images with similar patterns?

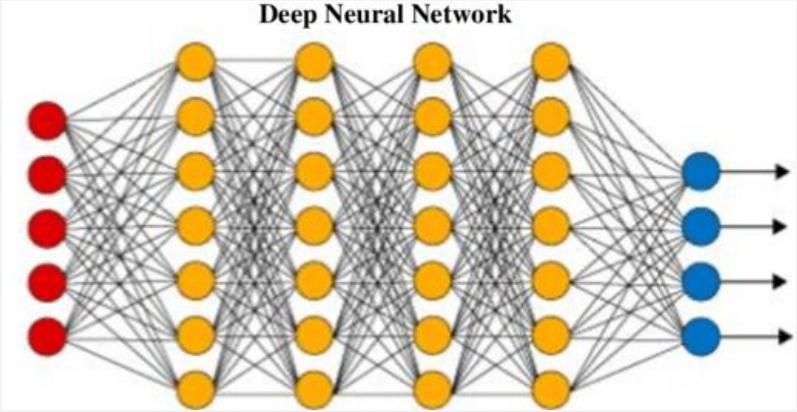
Neural Network doing Prediction



How many trainable parameters is in ChatGPT 3.5?

ChatGPT

ChatGPT 3.5 was based on the GPT 3.5 engine, which received training on over 175 billion training parameters. ChatGPT 4 took this training one step further, and is purportedly trained on over a trillion parameters.



AI, ML, DL, GenAI

Artificial Intelligence

Is the field of study that deals with creation of machine that exhibits animal or human intelligence.

Machine Learning

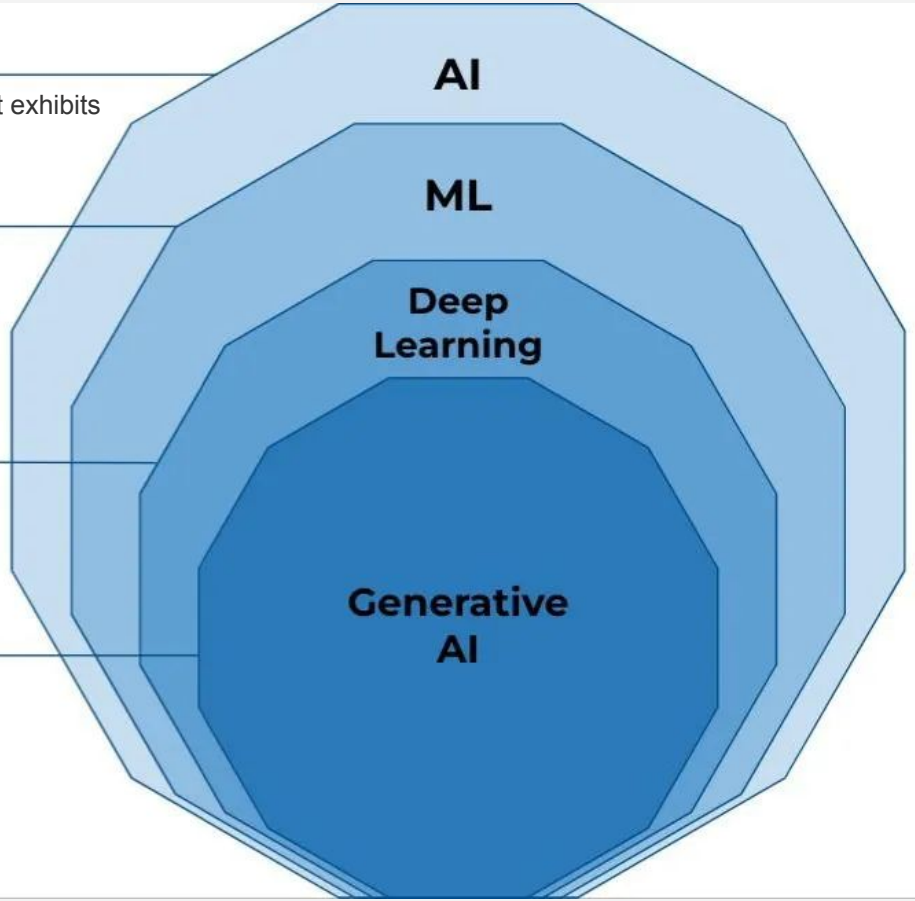
Is a branch of AI that focus on the creation of intelligent machines that learn from data. Another very well know branch inside AI is **Optimization**.

Deep Learning

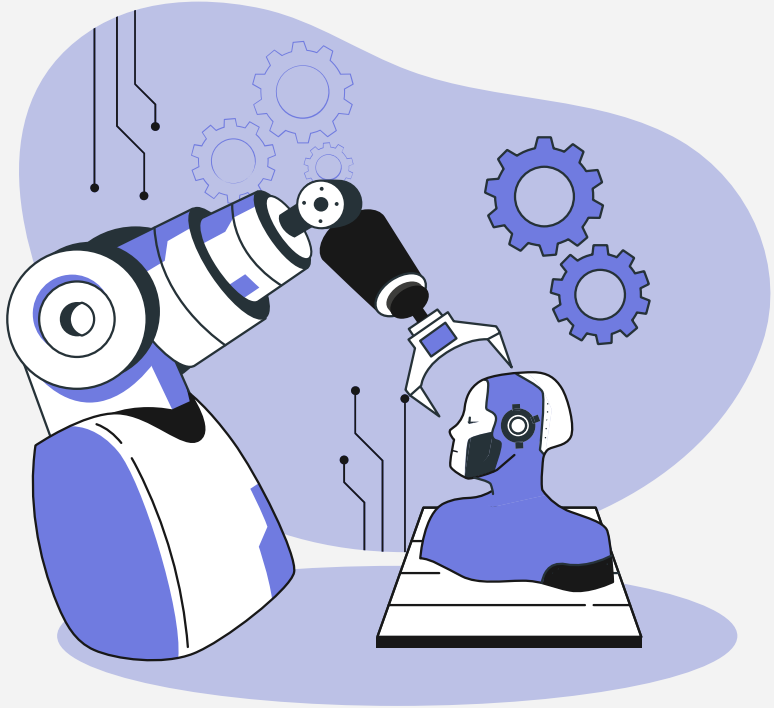
Is a subset of Machine Learning methods, based on **Artificial Neural Networks.**
Examples: CNNs, RNNs

Generative AI

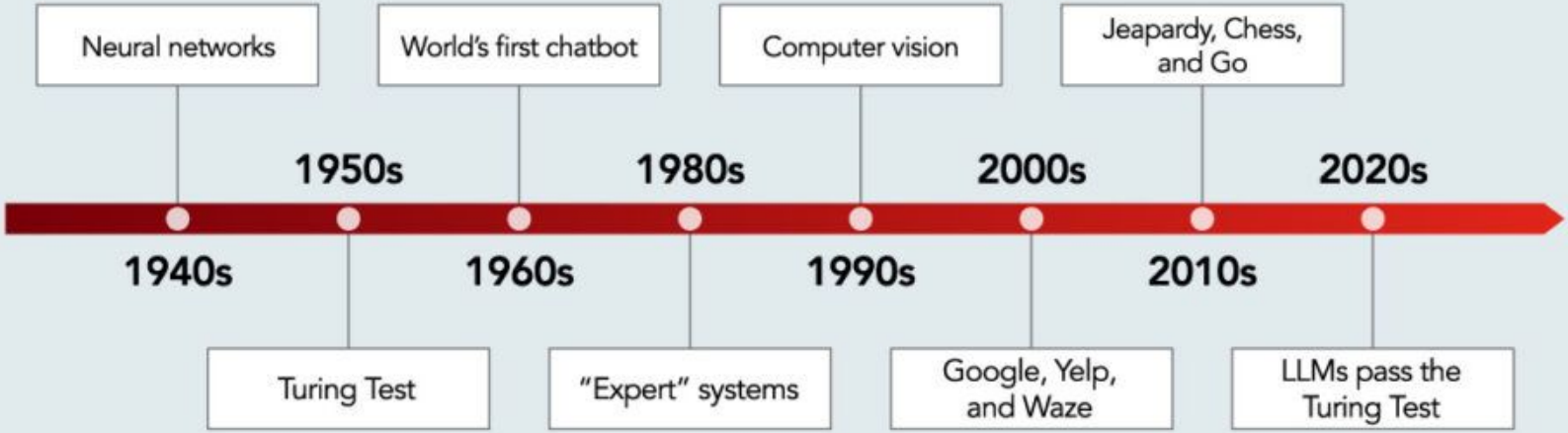
A type of ANNs that generate data that is similar to the data it was trained on.
Examples: GANs, LLMs



03
LLM

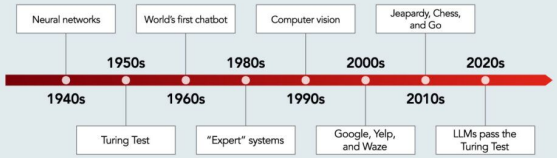


AI took 80+ years to become an overnight sensation



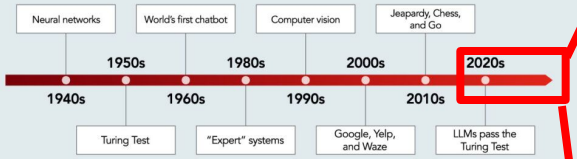
planview.

AI took 80+ years to become an overnight sensation



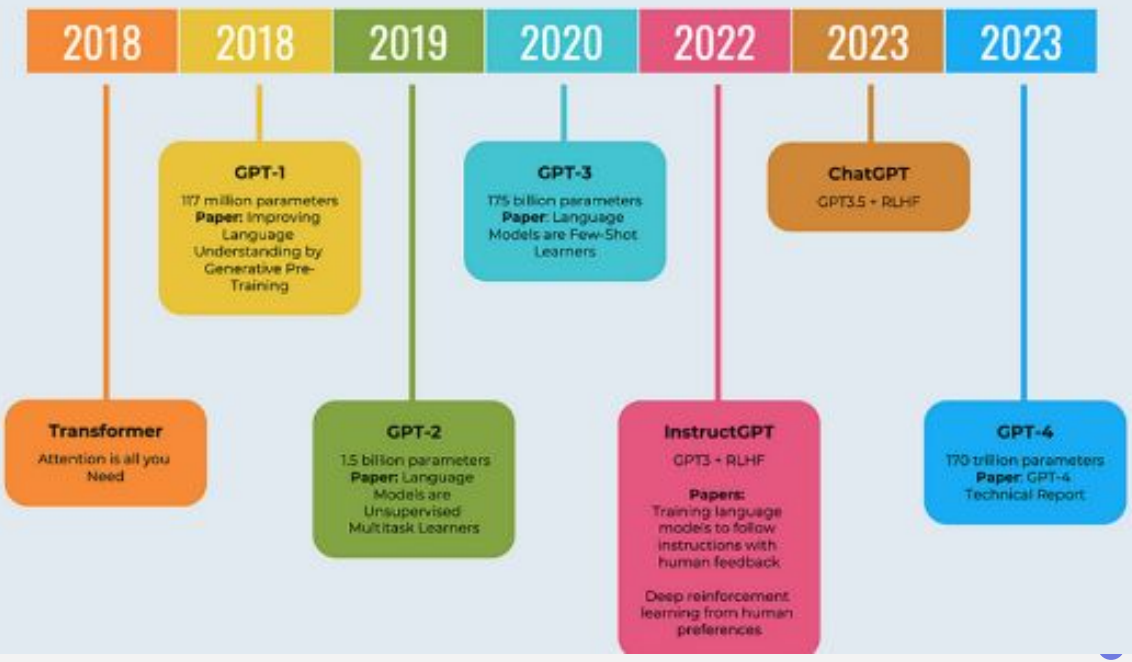
planview.

AI took 80+ years to become an overnight sensation



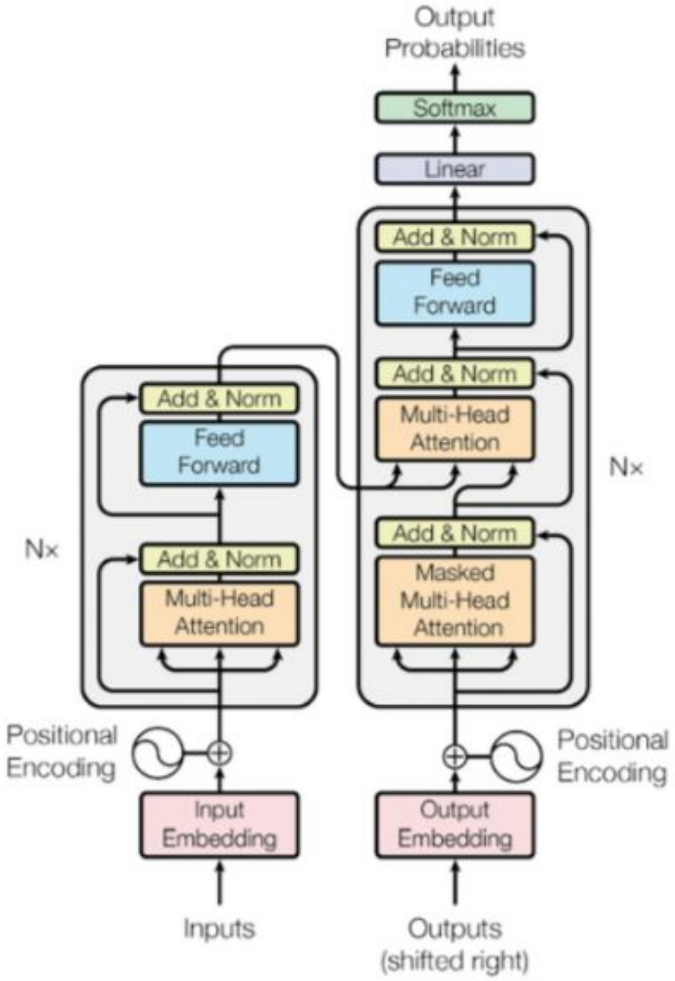
planview.

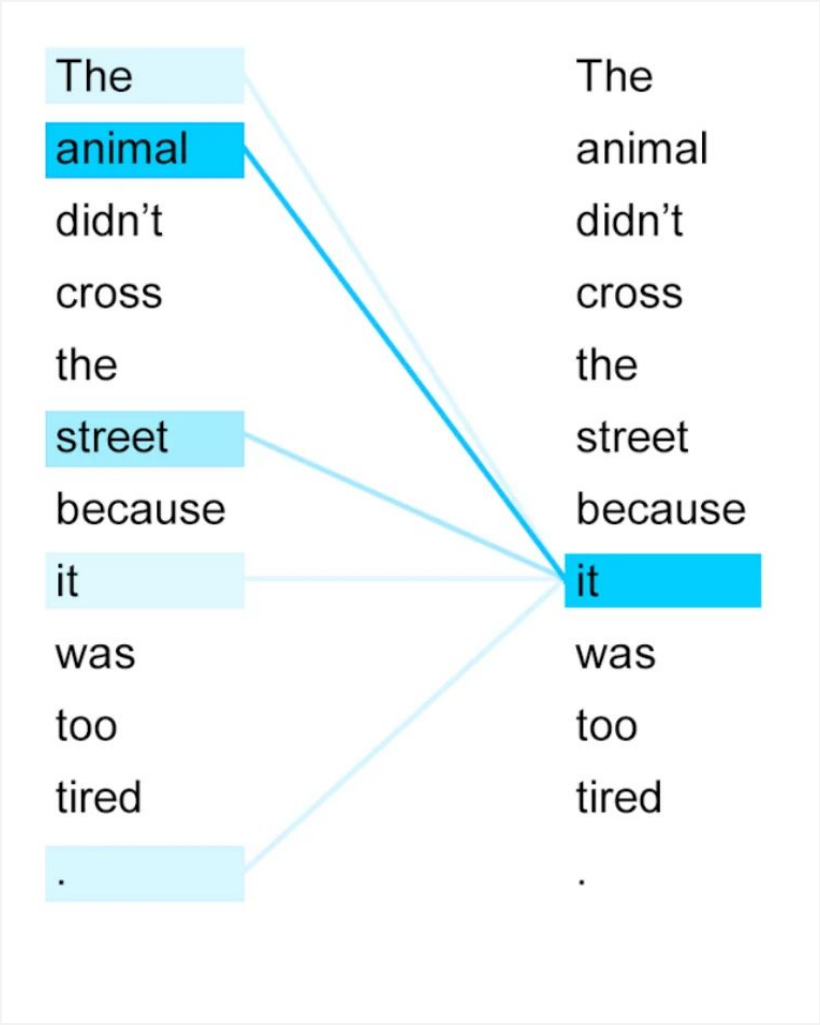
History of ChatGPT



Transformer

Attention Is All You Need





In transformers, **self-attention** allows the model to **weigh the importance of every word** in a sentence **relative to others** when making predictions.

In the sentence:
"The animal didn't cross the street because it was too tired."
— the word **"it"** is ambiguous, but most likely refers to **"the animal"** rather than **"the street."**

Using self-attention, the transformer can **focus more heavily on "animal"** when processing "it," because words like "tired" are more semantically associated with living things. The model learns these patterns from large text corpora, enabling it to resolve such references based on context.

This is milestone in NLP because it:

- Improves comprehension
- Supports human-like understanding

*previously, we’re just counting words (e.g. what’s the most frequent words, n-gram)

Why are LLMs an overnight sensation?

- **Language is a representation of human knowledge**
- Products of language is easy to generate (e.g. Internet, books, publications, messages, etc)
- Language is the best modality to train AI models to mimic human intelligence
- In the past year, LLMs have been successful in solving real-world problems
- **Best to think of LLMs as the compression with context of the digital human knowledge**

Large Language Models

Definition: LLMs are advanced AI models trained on vast amounts of text data to understand and generate human language.

Scale and Complexity: LLMs consist of billions to trillions of parameters, allowing them to handle complex language tasks like translation, summarization, and conversation.

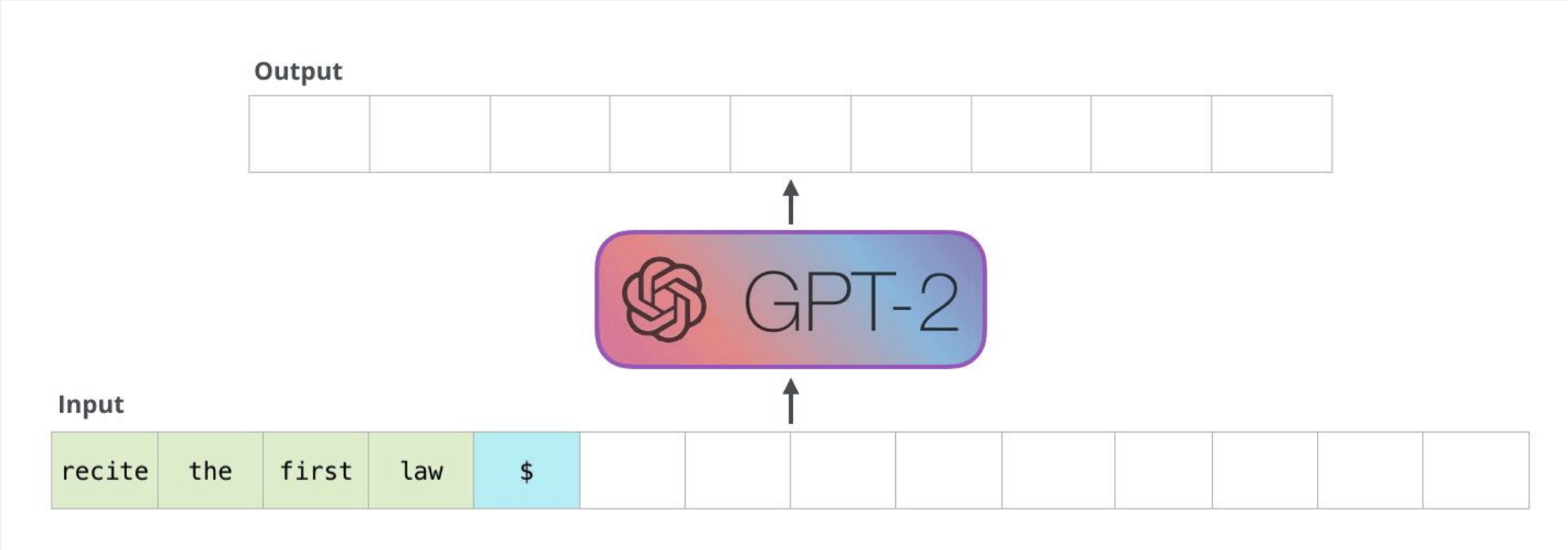
Learning Process: They learn patterns, context, and semantics from large datasets to generate coherent responses.

Real-World Applications: LLMs power chatbots, virtual assistants, content generation tools, code-writing systems, and more across industries.

Challenges: Despite their capabilities, LLMs face challenges like bias in training data, high computational costs, and sometimes generating inaccurate or nonsensical responses.

How does LLM work?

- Use of autoregression



How does LLM work?

- Use of autoregression to predict the next word (token)

$$p(x_1, x_2, \dots, x_L) = p(x_1)p(x_2|x_1)\cdots p(x_L|x_1, x_2 \cdots x_{L-1}) = \prod_{i=1}^{L} p(x_i|x_{1:i-1})$$

$$p(the, cat, sat, on, the, mat) =$$
$$p(the) \cdot$$
$$p(cat|the) \cdot$$
$$p(sat|the, cat) \cdot$$
$$p(on|the, cat, sat) \cdot$$
$$p(the|the, cat, sat, on) \cdot$$
$$p(mat|the, cat, sat, on, the)$$

Capabilities

- **Question Answering**
- Word Analogies
- Summarization
- Language Translation
- In-Context Learning

The capital of Malaysia is _____

Capabilities

- Question Answering
- **Word Analogies**
- Summarization
- Language Translation
- In-Context Learning

Capabilities

- Question Answering
- Word Analogies
- **Summarization**
- Language Translation
- In-Context Learning

The Olympic Games is a major international multi-sport event. During its history, both the Summer and Winter Games were a subject of many scandals, controversies, and boycotts. Some Olympic Games were not conducted due to major wars - such as World War I and World War II. The Games originated in Ancient Greece, and now are conducted every four years by the International Olympic Committee. Athletes from more than 200 nations participate in these games. The symbol of the Olympic Games - five interlocking colored rings, representing the continents of the world united by Olympism. The goal of the Olympic Movement is to contribute to building a peaceful and better world by educating youth through sport practiced without discrimination of any kind and in the Olympic spirit.

Summary

The Olympic Games, originating in Ancient Greece, are a major international event occurring every four years, involving over 200 nations. The event, overseen by the International Olympic Committee, has faced scandals, controversies, and were cancelled due to world wars. The five interlocked rings symbolize the unity among continents, promoting peace and youth education through non-discriminatory sport.

Capabilities

- Question Answering
- Word Analogies
- Summarization
- **Language Translation**
- In-Context Learning

Once upon a time, a quick rabbit named Ralph lived in a lush forest alongside a wise old turtle, Thomas. Ralph, full of energy and spirit, often teased the slow-moving Thomas, eventually challenging him to a race. Confident of his victory, Ralph sped ahead from the starting line, swiftly leaving Thomas far behind. Deciding to take a quick nap under a shady tree, Ralph thought, "Even with my rest, I'll surely win."

On the other hand, Thomas consistently trudged forward, maintaining his slow, steady pace. As the afternoon transitioned into evening, Ralph woke, alarmed to see that the sun was setting. Quickly, Ralph raced towards the finish line, but as he rounded the bend, he saw Thomas already there, smiling wisely.

Baffled and humbled, Ralph approached the finish line, comprehending the invaluable lesson of Thomas's victory: consistency over speed. The race brought newfound respect in Ralph's heart for Thomas, embedding a lifelong friendship between the two. The quick rabbit learned the wisdom embedded in Thomas's slow and steady philosophy that day.

Noong isang panahon, may mabilis na kuneho na nagngangalang Ralph na nakatira sa isang saganang gubat kasama ng isang marunong na matandang pagong, na si Thomas. Madalas asarin ni Ralph, na puno ng enerhiya at sigla, ang mabagal na si Thomas, at sa wakas ay hinamon nya ito sa isang karera. Sigurado sa kanyang tagumpay, si Ralph ay agad na umarangkada mula sa simula, mabilis na iniwanan si Thomas sa malayo.

Nagpasya siyang magpahinga at matulog sa ilalim ng isang puno, iniisip na "Kahit na magpahinga pa ako, sigurado akong ako ang mananalo."

Sa kabilang banda, si Thomas ay tuloy-tuloy na nagpatuloy, sinusunod ang kanyang mabagal ngunit walang patid na tulin. Nang maghapon na naglipat sa gabi, si Ralph ay nagising, nababahala sa paglubog ng araw. Mabilis, dumaloy si Ralph patungo sa finish line, ngunit nang siya ay nag-ikot, nakita niya na si Thomas ay nandoon na, ngumiti ng marunong.

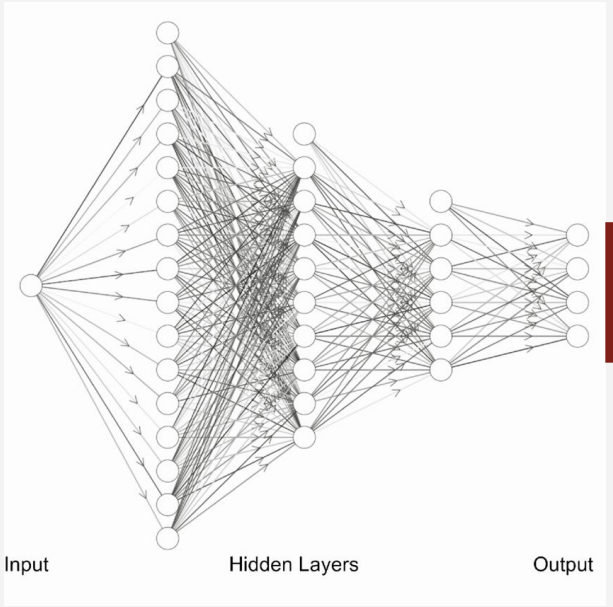
Nalilito at nababaon, lumapit si Ralph sa finish line, nauunawaan ang mahalagang aral ng tagumpay ni Thomas: ang konsistensya ay higit pa sa bilis. Ang karera ay nagdala ng bagong pag-galang sa puso ni Ralph para kay Thomas, nagtatanim ng dalisay na pagkakaibigan sa pagitan ng dalawa. Natutunan ng mabilis na kuneho ang karunungan na nakabaon sa mabagal at patuloy na pilosopiya ni Thomas noong araw na iyon.

Capabilities

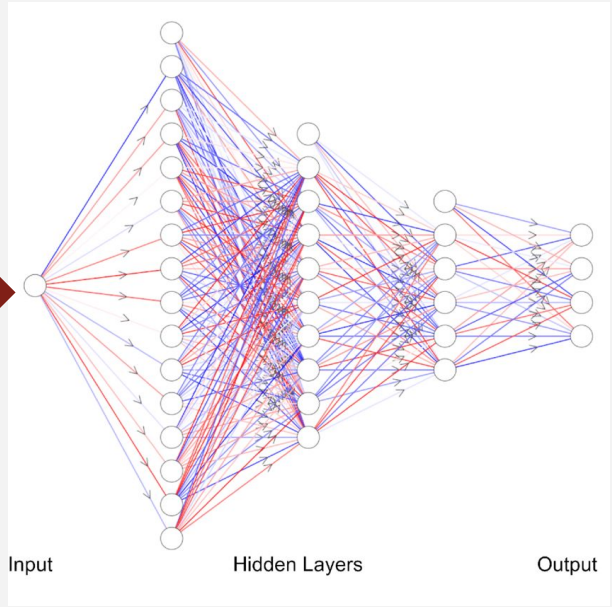
- Question Answering
- Word Analogies
- Summarization
- Language Translation
- **In-Context Learning**

```
# Chatbot - GPT-3 model

chatbot = GPT3() # User initiates a conversation

user_says = "Hello. I want to talk about cars."

chatbot_response = chatbot.respond(user_says)

print(chatbot_response) # "Sure, what would you like to know about cars?"

user_says = "What's the fastest car in the world?"

chatbot_response = chatbot.respond(user_says)

print(chatbot_response) # "As of my training data up until September 2021, the fastest car in the world is the Bugatti Chiron Super Sport 300+ with a top speed of 304.77 mph."

user_says = "What's its engine power?"

chatbot_response = chatbot.respond(user_says) print(chatbot_response) # "The Bugatti Chiron Super Sport 300+ has a quad-turbocharged 8.0 litre W16 engine that produces 1578 horsepower."
```

Training the LLM...



MODEL TRAINING



Training the LLM...

Pre-Training

Pre-Training Data

- **Publicly Available Data**: Books, encyclopedias (e.g., Wikipedia), open-access articles, forums, and websites.
- **Licensed Content**: Data acquired through proper agreements for enhanced quality.
- **Code Repositories**: Publicly accessible coding examples from platforms like GitHub and Stack Overflow.

BASE MODEL

Training the LLM...

Pre-Training

Pre-Training Data

- **Publicly Available Data**: Books, encyclopedias (e.g., Wikipedia), open-access articles, forums, and websites.
- **Licensed Content**: Data acquired through proper agreements for enhanced quality.
- **Code Repositories**: Publicly accessible coding examples from platforms like GitHub and Stack Overflow.

BASE MODEL

Trivia:

GPT in ChatGPT stands for **G**enerative **P**re-trained **T**ransformer.

Training the LLM...

Pre-Training → Fine-Tuning

Pre-Training Data

- **Publicly Available Data**: Books, encyclopedias (e.g., Wikipedia), open-access articles, forums, and websites.
- **Licensed Content**: Data acquired through proper agreements for enhanced quality.
- **Code Repositories**: Publicly accessible coding examples from platforms like GitHub and Stack Overflow.

Fine-Tuning Data

- Can we further train the base model (aka pre-trained model)?

BASE MODEL → FINE-TUNED MODEL

Training the LLM...

Pre-Training → Fine-Tuning → Instruction-Tuning

Pre-Training Data

- **Publicly Available Data**: Books, encyclopedias (e.g., Wikipedia), open-access articles, forums, and websites.
- **Licensed Content**: Data acquired through proper agreements for enhanced quality.
- **Code Repositories**: Publicly accessible coding examples from platforms like GitHub and Stack Overflow.

Fine-Tuning Data

- Can we further train the base model (aka pre-trained model)?

Instruction-Tuning Data

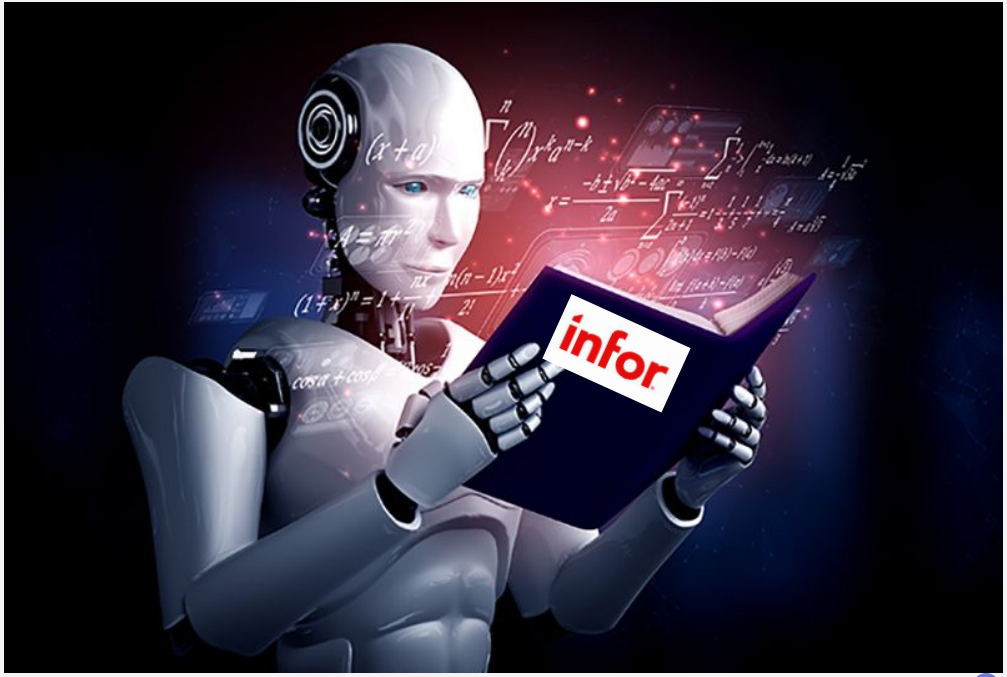
- **Instructional Fine-Tuning**: Curated datasets to improve response relevance and alignment to user instructions.

sample:

System prompts: “You are a helpful assistant for an ERP System. Answer queries related to infor system only. ”

BASE MODEL → FINE-TUNED MODEL → ALIGNED MODEL

"I see, AI can utilize existing documents to expand its knowledge base."

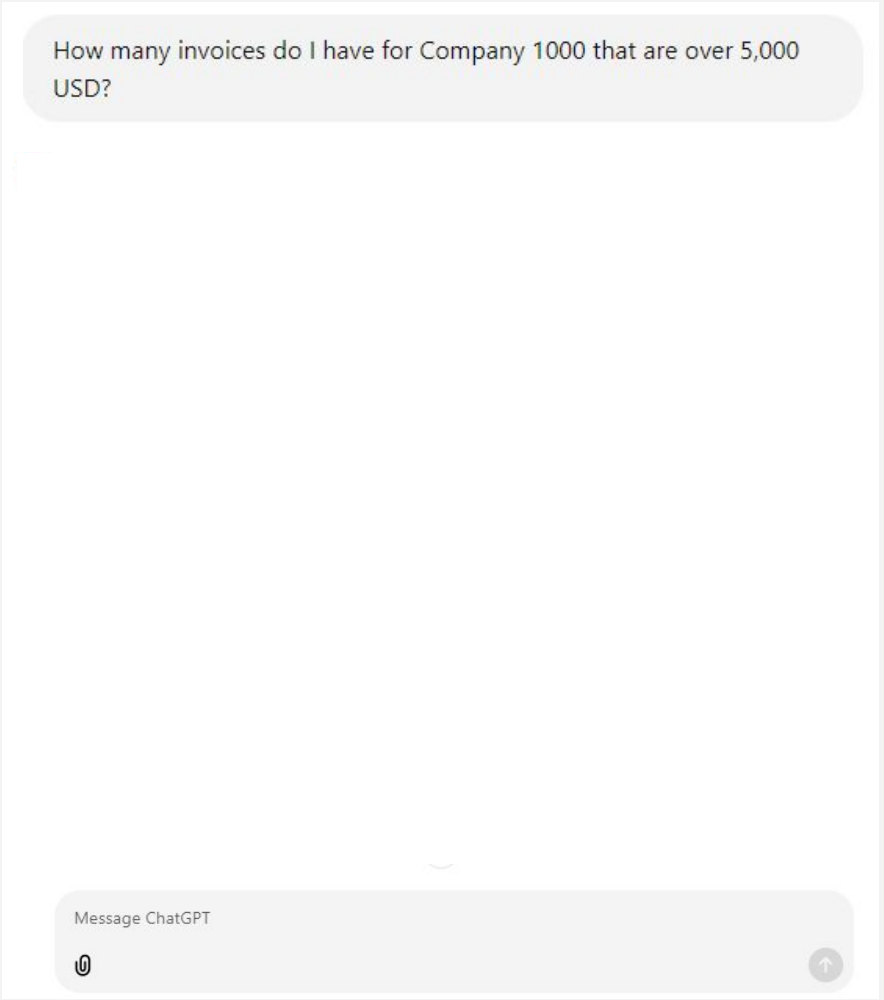


“But how can AI access real-time data?”

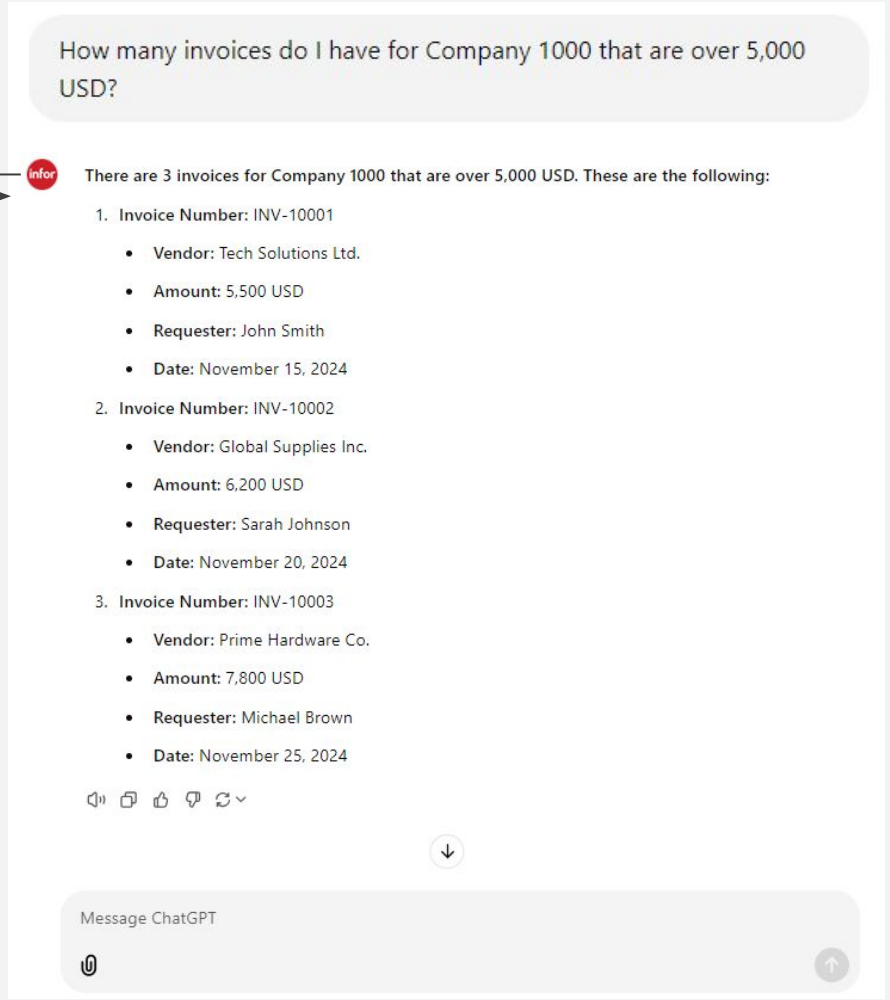
Use RAG

- What is RAG?
  - Retrieval-Augmented Generation
  - A method combining **retrieval systems** and **generative models** (e.g., LLMs).
- Designed to enhance generative AI by incorporating **external knowledge** during response generation.

RAG in Infor FSM?



RAG in Infor FSM?



- A method combining **retrieval systems** and **generative models** (e.g., LLMs).
- Designed to enhance generative AI by incorporating **external knowledge** during response generation.

Possible Use Cases of RAG in an ERP system

Financials

- **Automated Financial Insights**:
  Retrieve real-time financial data for instant analysis and report generation.

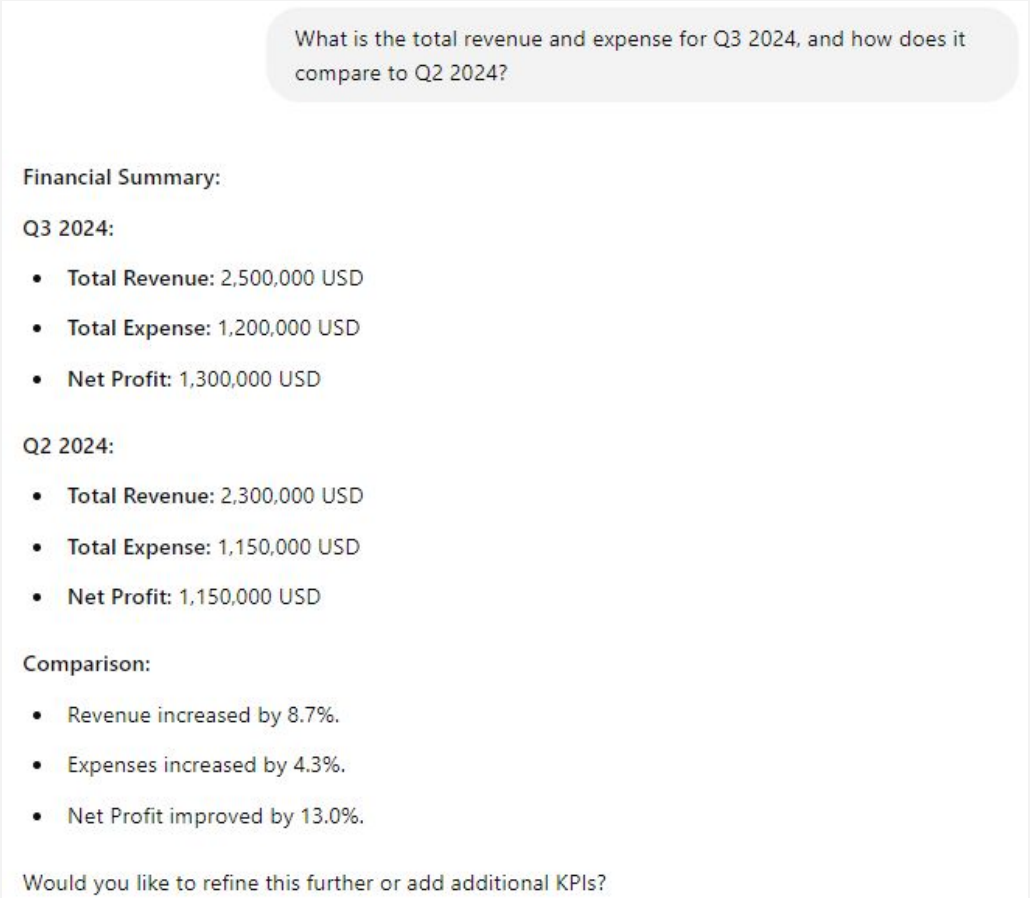
  Sample Query: "What is the total revenue and expense for Q3 2024, and how does it compare to Q2 2024?"

- **Audit Assistance**:
  Answer audit queries by pulling relevant records from past transactions.

  Sample Query: "Provide all transactions made by Vendor XYZ in Q3 2024 exceeding 10,000 USD, including payment dates and approvers."

- **Expense Optimization**:
  Identify patterns and recommend cost-saving measures using historical spend data.

  Sample: "What are the top three categories with the highest expenses over the past six months, and what cost-saving measures can be recommended?"



Possible Use Cases of RAG in an ERP system

Financials

- **Automated Financial Insights**:
  Retrieve real-time financial data for instant analysis and report generation.

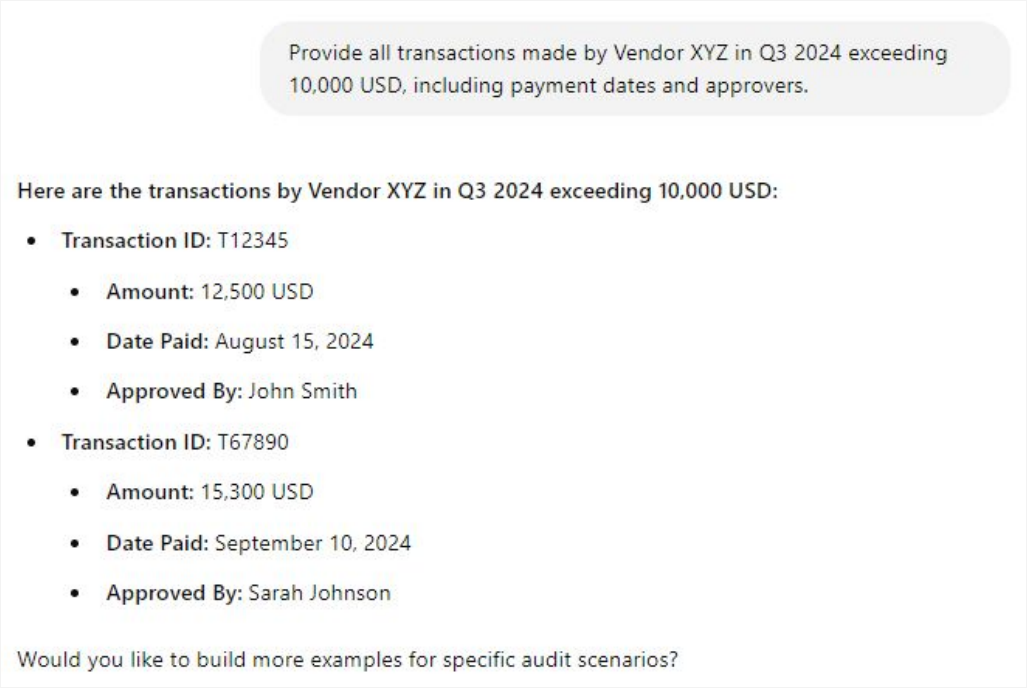
  Sample Query: "What is the total revenue and expense for Q3 2024, and how does it compare to Q2 2024?"

- **Audit Assistance**:
  Answer audit queries by pulling relevant records from past transactions.

  Sample Query: "Provide all transactions made by Vendor XYZ in Q3 2024 exceeding 10,000 USD, including payment dates and approvers."

- **Expense Optimization**:
  Identify patterns and recommend cost-saving measures using historical spend data.

  Sample: "What are the top three categories with the highest expenses over the past six months, and what cost-saving measures can be recommended?"



Possible Use Cases of RAG in an ERP system

Financials

- **Automated Financial Insights**:
  Retrieve real-time financial data for instant analysis and report generation.

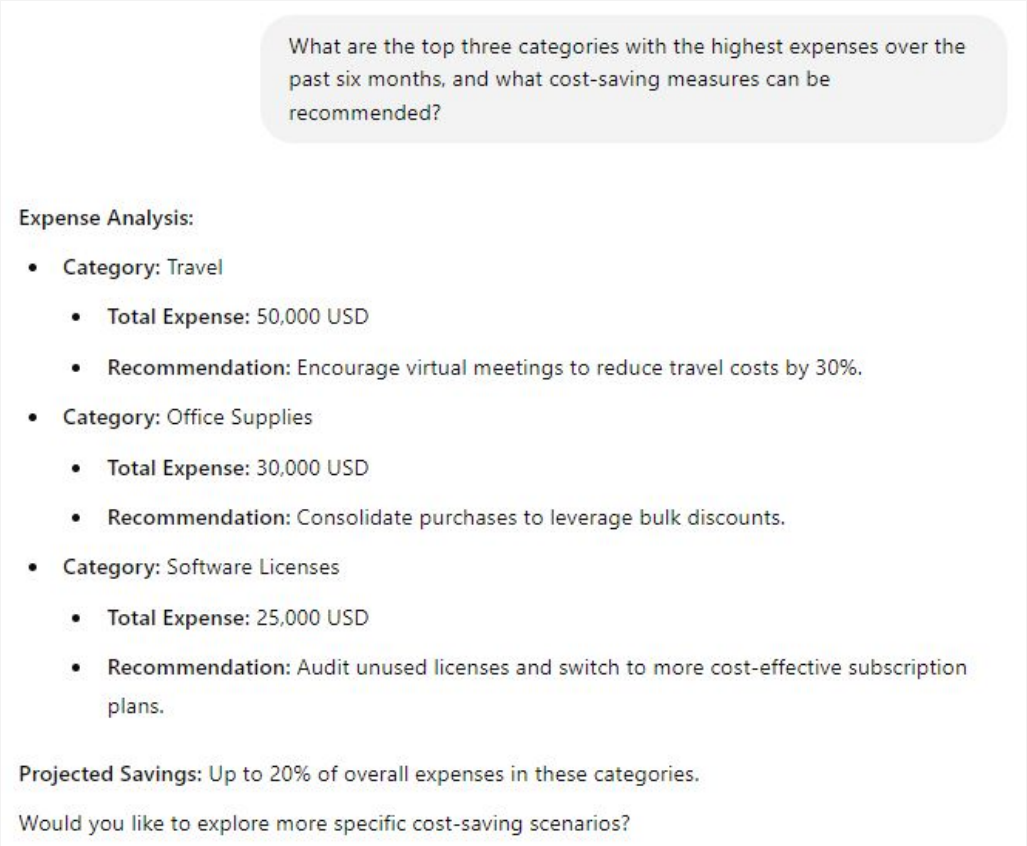
  Sample Query: "What is the total revenue and expense for Q3 2024, and how does it compare to Q2 2024?"

- **Audit Assistance**:
  Answer audit queries by pulling relevant records from past transactions.

  Sample Query: "Provide all transactions made by Vendor XYZ in Q3 2024 exceeding 10,000 USD, including payment dates and approvers."

- **Expense Optimization**:
  Identify patterns and recommend cost-saving measures using historical spend data.

  Sample: "What are the top three categories with the highest expenses over the past six months, and what cost-saving measures can be recommended?"



Possible Use Cases of RAG in an ERP system

Supply Chain

- **Demand Forecasting**:
  Retrieve historical sales data to generate accurate demand predictions.

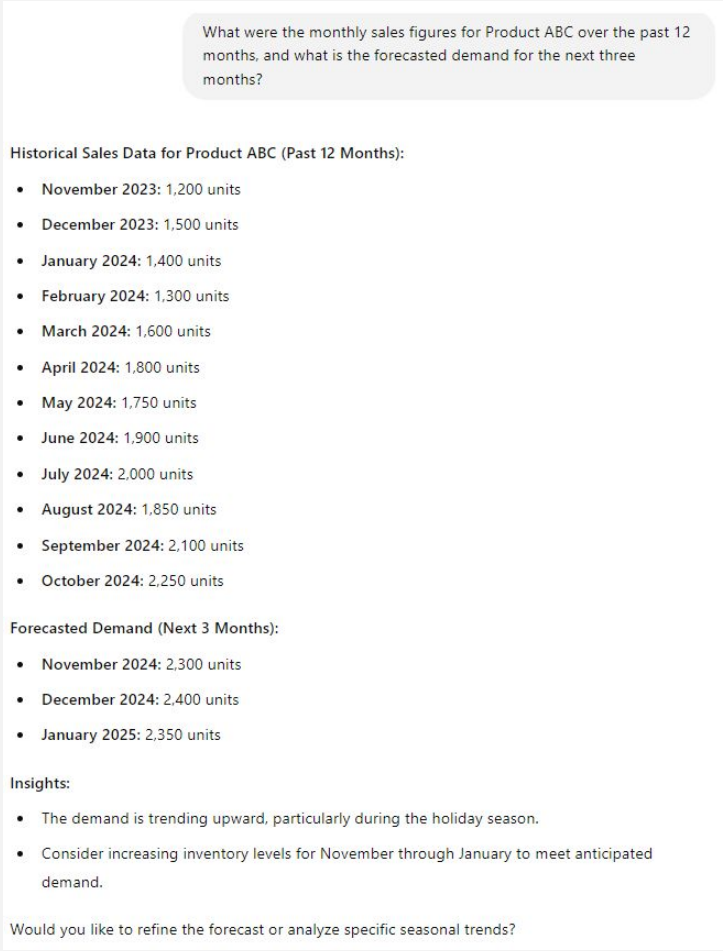
  Sample: “What were the monthly sales figures for Product ABC over the past 12 months, and what is the forecasted demand for the next three months?”

- **Supplier Insights**:
  Summarize supplier performance and contract terms from past records.

  Sample: “Provide a summary of Supplier XYZ's performance and contract terms for the past 12 months.”

- **Real-Time Query Handling**:
  Fetch shipment status, inventory levels, or delivery schedules dynamically.

  Sample: “What is the current status of the shipment for Order #56789, and what are the available inventory levels for Product ABC?”



Possible Use Cases of RAG in an ERP system

Supply Chain

- **Demand Forecasting**:
  Retrieve historical sales data to generate accurate demand predictions.

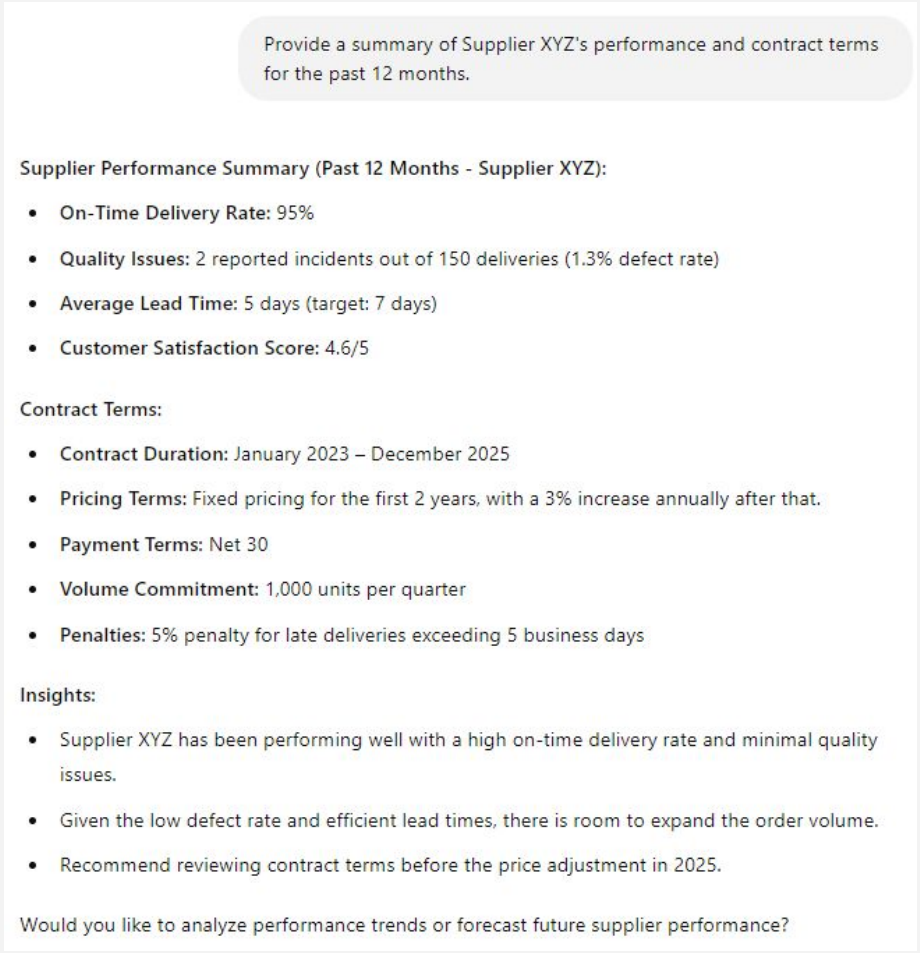
  Sample: “What were the monthly sales figures for Product ABC over the past 12 months, and what is the forecasted demand for the next three months?”

- **Supplier Insights**:
  Summarize supplier performance and contract terms from past records.

  Sample: “Provide a summary of Supplier XYZ's performance and contract terms for the past 12 months.”

- **Real-Time Query Handling**:
  Fetch shipment status, inventory levels, or delivery schedules dynamically.

  Sample: “What is the current status of the shipment for Order #56789, and what are the available inventory levels for Product ABC?”



Possible Use Cases of RAG in an ERP system

Supply Chain

- **Demand Forecasting**:
  Retrieve historical sales data to generate accurate demand predictions.

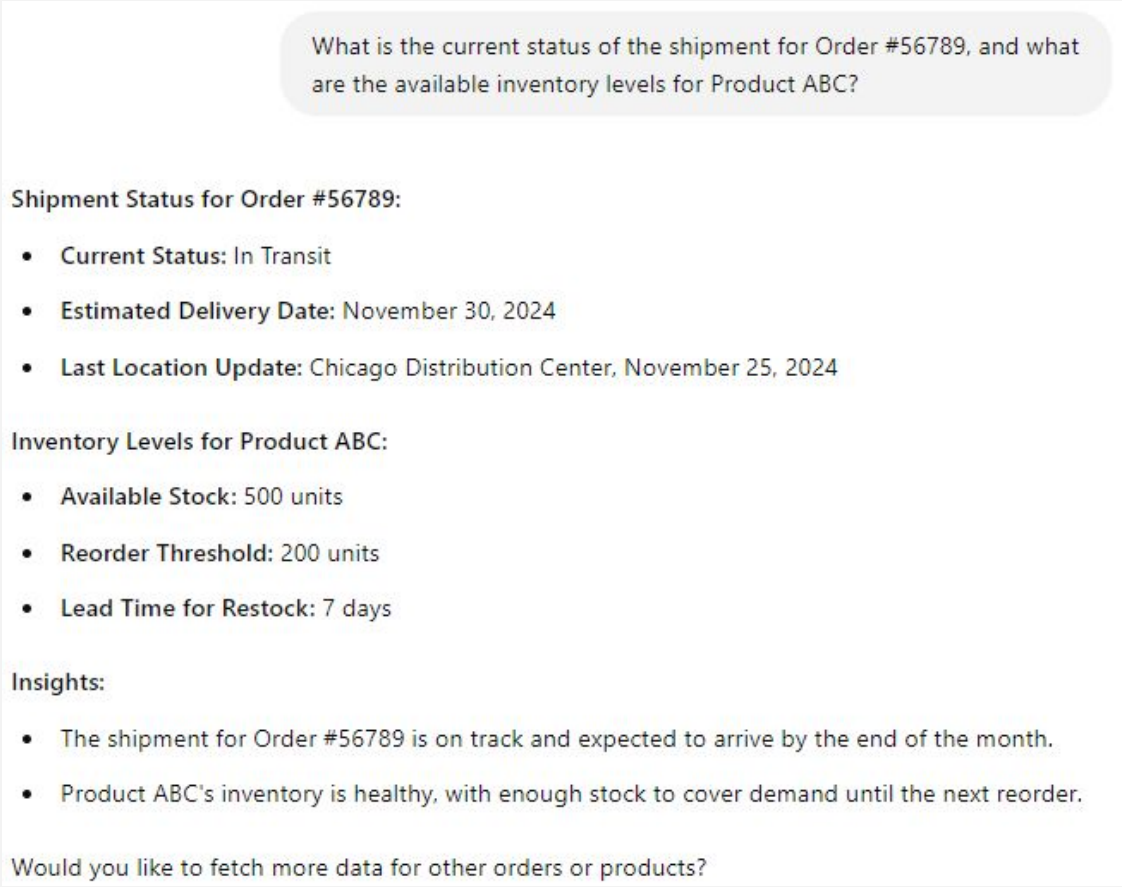
  Sample: “What were the monthly sales figures for Product ABC over the past 12 months, and what is the forecasted demand for the next three months?”

- **Supplier Insights**:
  Summarize supplier performance and contract terms from past records.

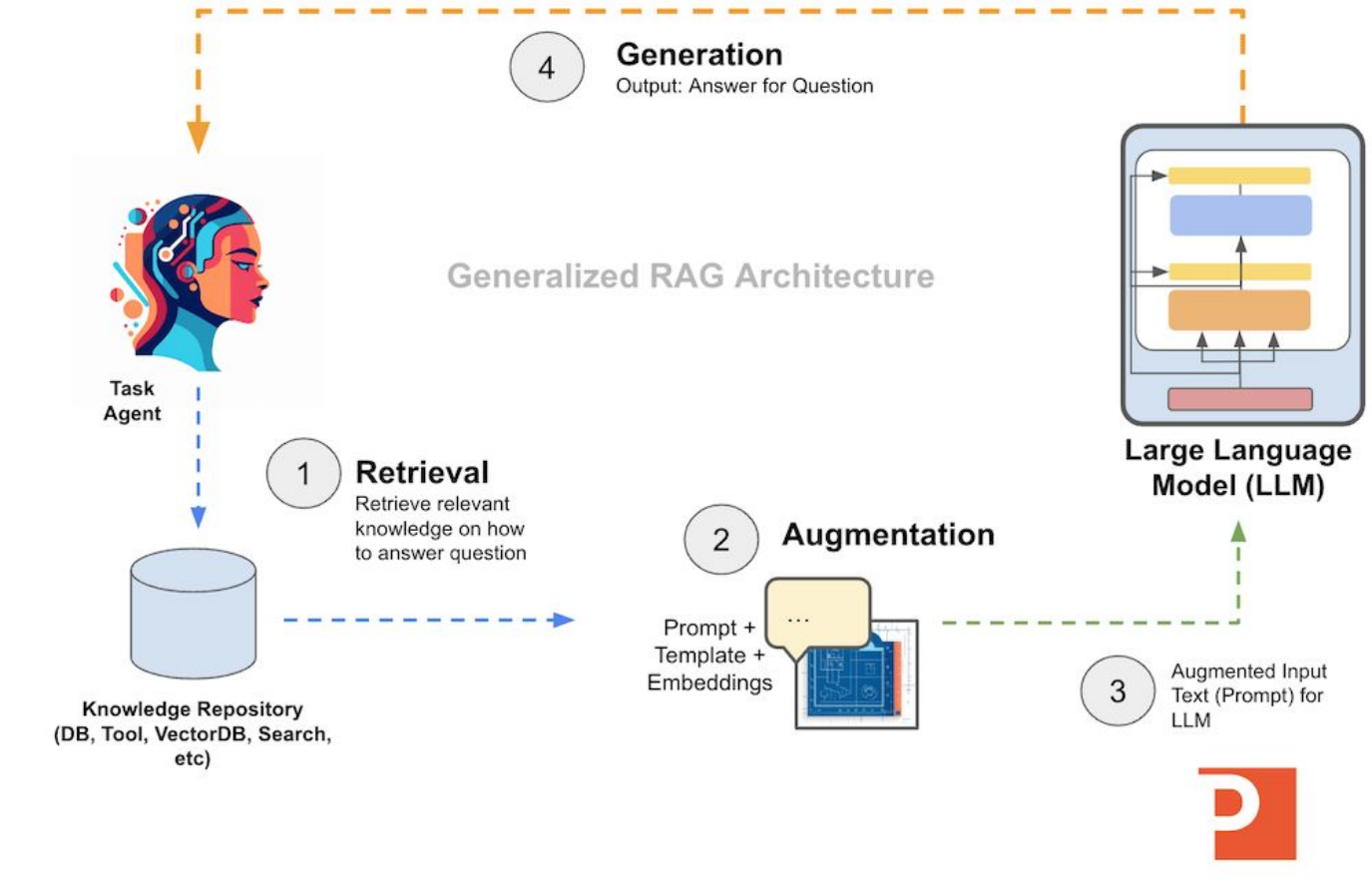
  Sample: “Provide a summary of Supplier XYZ's performance and contract terms for the past 12 months.”

- **Real-Time Query Handling**:
  Fetch shipment status, inventory levels, or delivery schedules dynamically.

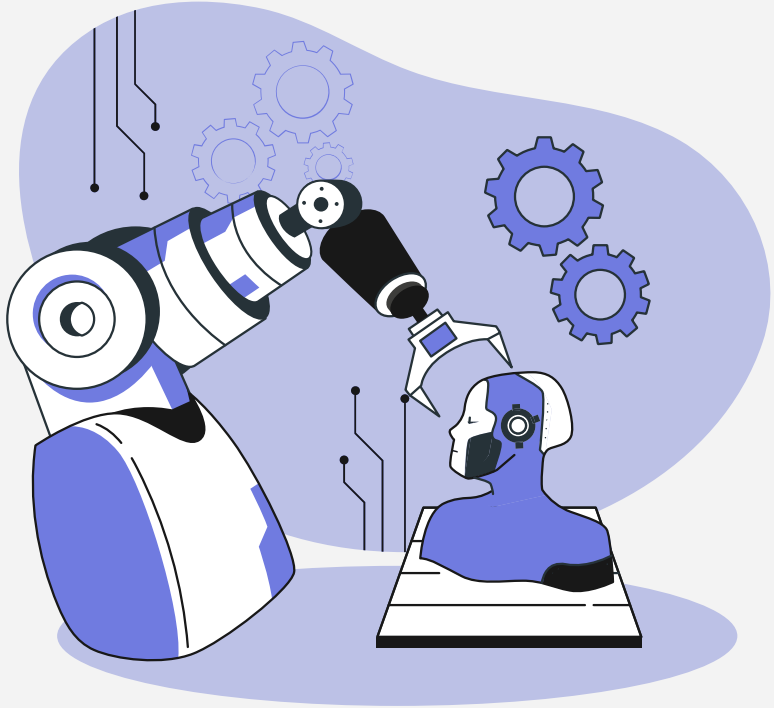
  Sample: “What is the current status of the shipment for Order #56789, and what are the available inventory levels for Product ABC?”



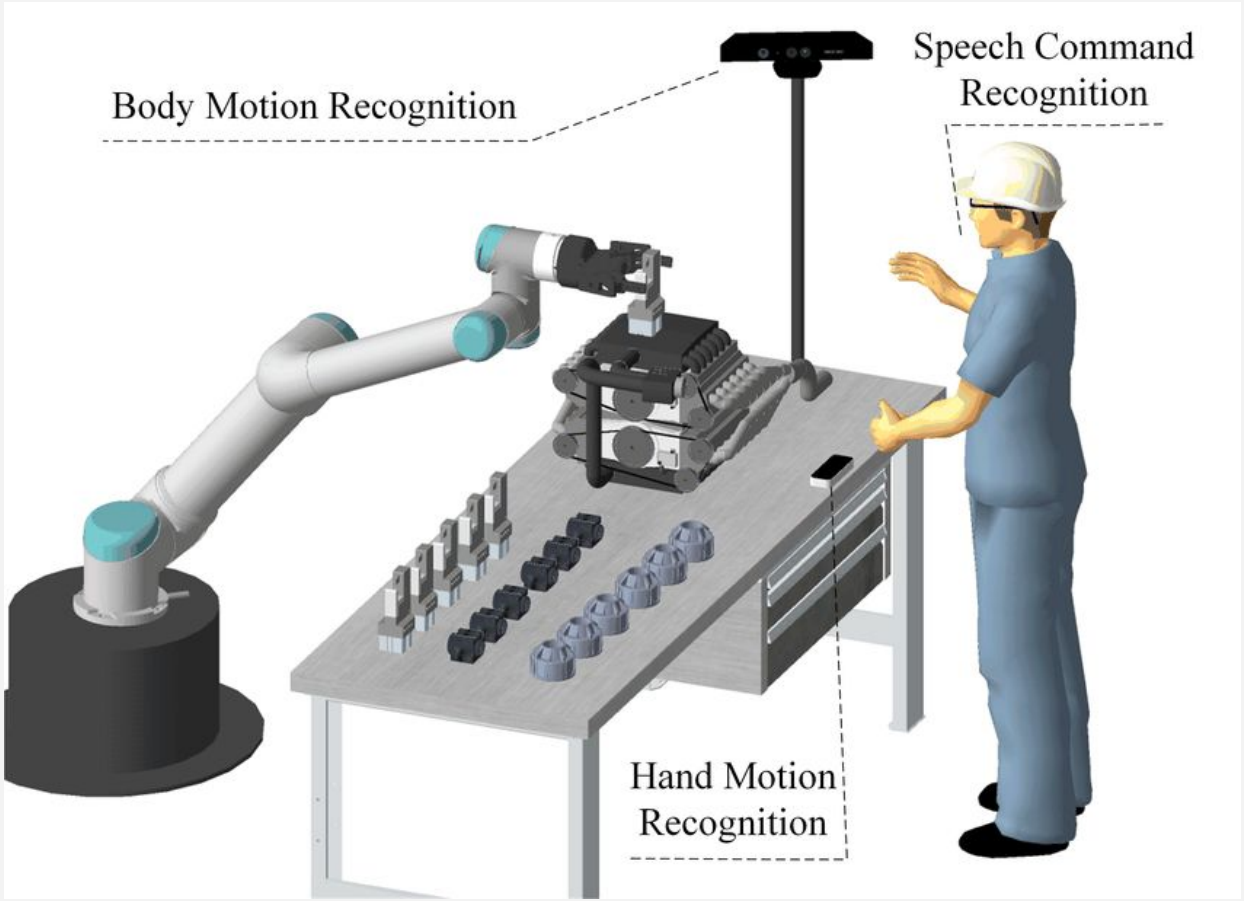
RAG + LLM


4
Generation
Output: Answer for Question
Generalized RAG Architecture
Task Agent
1
Retrieval
Retrieve relevant knowledge on how to answer question
Knowledge Repository (DB, Tool, VectorDB, Search, etc)
2
Augmentation
Prompt + Template + Embeddings
3
Augmented Input Text (Prompt) for LLM
Large Language Model (LLM)

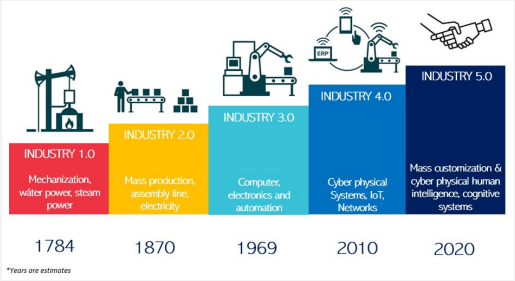
04
Agentic AI



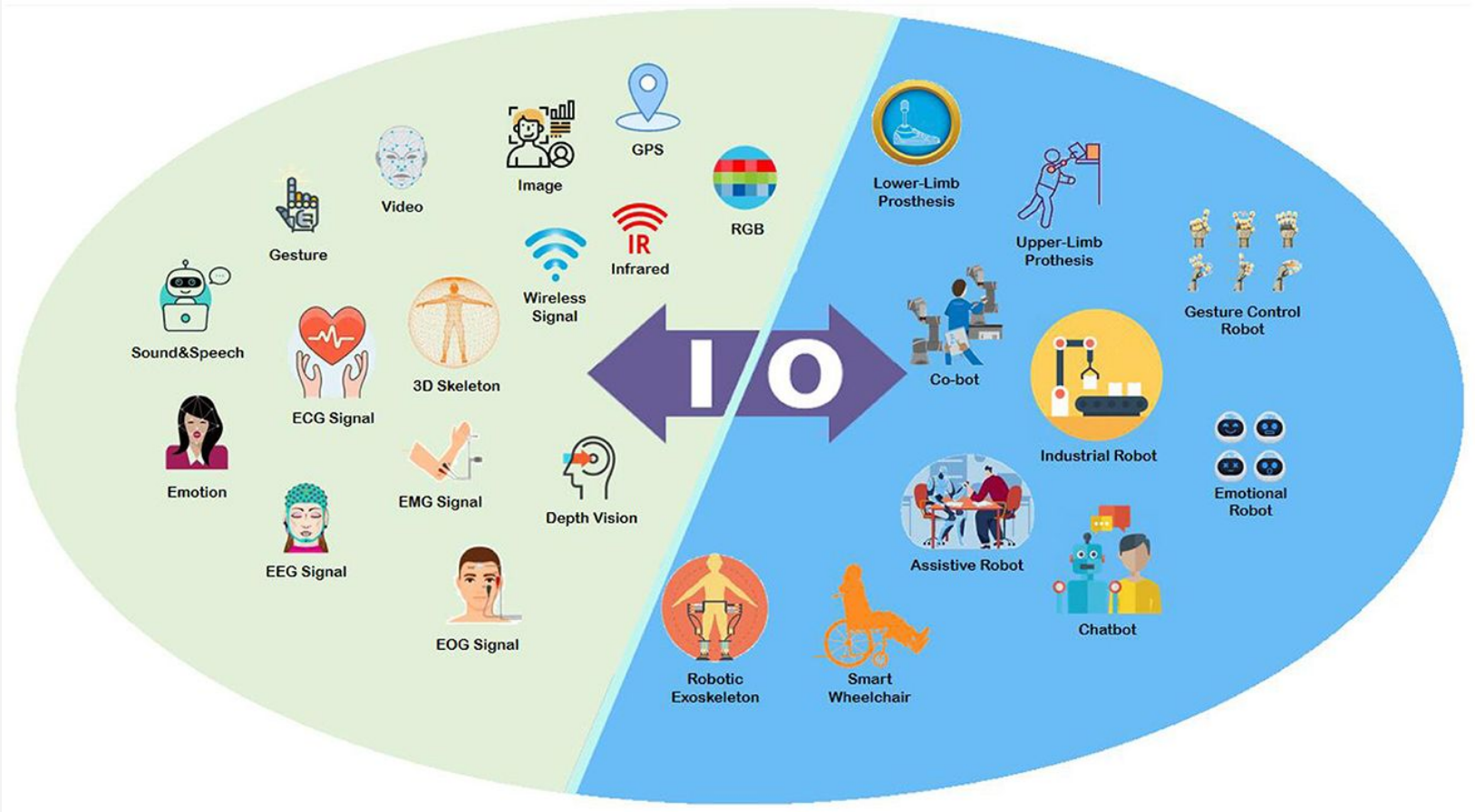
Multimodal Systems



One of the promises of Industry 5.0 is the use of co-bots, enabling seamless interaction between humans and robots.

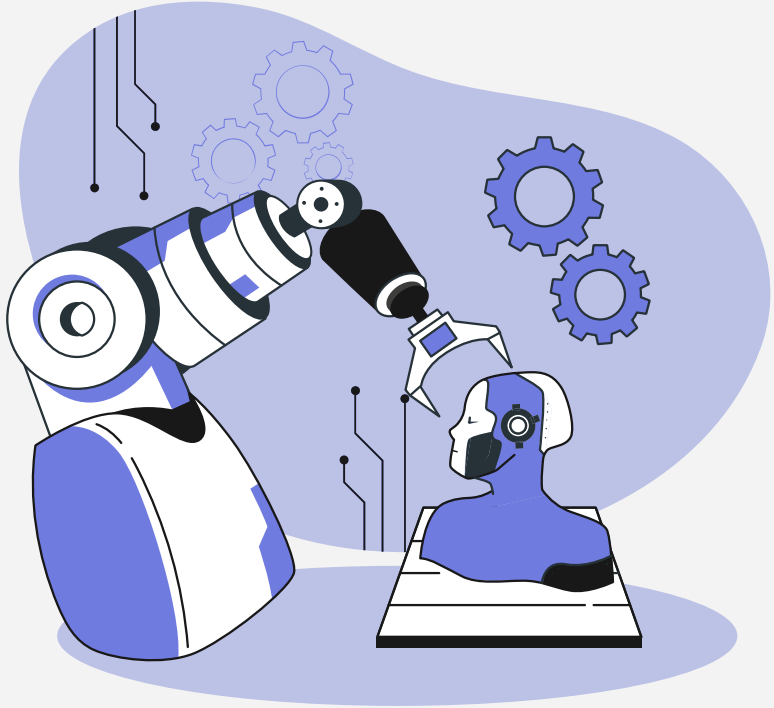


Multimodal Systems




Astribot S1
Hello World!

05
Q&A



Activity 1 - Build your own use case

1. Think of a use case that you want to finetune Deepseek with. Some ideas below:

Filipino Idiom & Proverb Explainer

Train DeepSeek to explain and translate Filipino sayings and proverbs with step-by-step logic.
📘 Example: *"What does 'Aanhin pa ang damo kung patay na ang kabayo?' mean?"*

Cultural Context Interpreter

Enable the model to explain local humor, references, or expressions like *'Jejemon'*, *'Jowa'*, *'Lodi'*, etc., in proper English with cultural insight.
📘 Example: *"What does 'Petmalu' mean and when is it used?"*

Workout Plan Generator Based on Goal

Train the model to provide beginner-friendly fitness plans depending on a user's body goals.
📘 Example: I want to build muscle but lose belly fat.

Activity 1 - Build your own use case

2. Ask ChatGPT to generate your dataset

📝 **Sample Prompt for Generating a CoT Dataset**

Prompt (for the model or instruction to students):

You are helping create a dataset to fine-tune deepseek. For each entry, provide three components:

1. **Question** — A concise, clear question or task. The first question must be:
   Who is <insert your name>?
   Then, based on what you know about me, generate the subsequent questions.
2. **Chain of Thought (CoT)** — A detailed, step-by-step reasoning process that explains how you arrive at the answer. This "thinking aloud" helps the model learn to reason before answering.
3. **Response** — A clear, direct final answer to the question.

Focus on questions that require reasoning, explanation, or synthesis, rather than simple factual recall.

Begin with the required first question, then generate ten additional questions with their respective CoT and Response based on your knowledge of me. Create 3 columns: Question - CoT - Response and then output a table

Activity 1 - Build your own use case

3. Create a Github repository named **AIFirst_Week2_[Surname]**, upload the dataset in the repository with the name **data_[topic]_[Surname].csv**. Submit the github link in the submissions channel.

Activity 2 - Finetune Deepseek

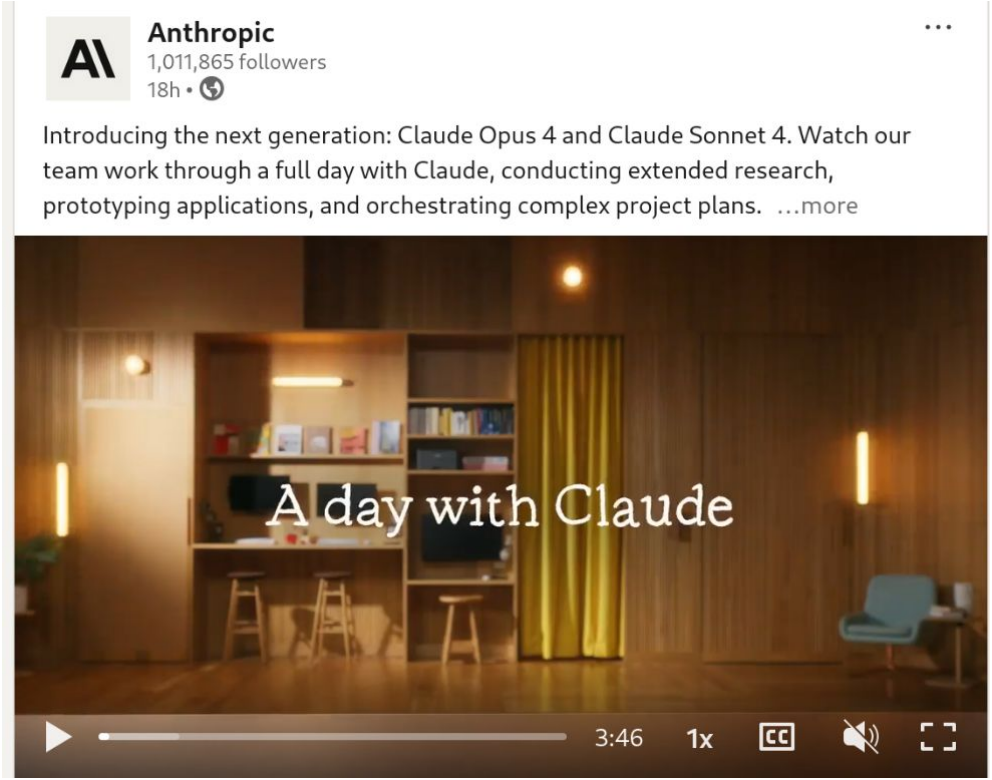
1. Download a copy of the Finetuning notebook from Github
2. Open Google Colab
3. Upload the notebook
4. Rename your notebook to “Finetuning_Week2_Surname”
5. Upload your generated dataset from Activity 1
6. Run the notebook
7. Interpret and observe the results
8. Upload your notebook to the github repository from Activity 1

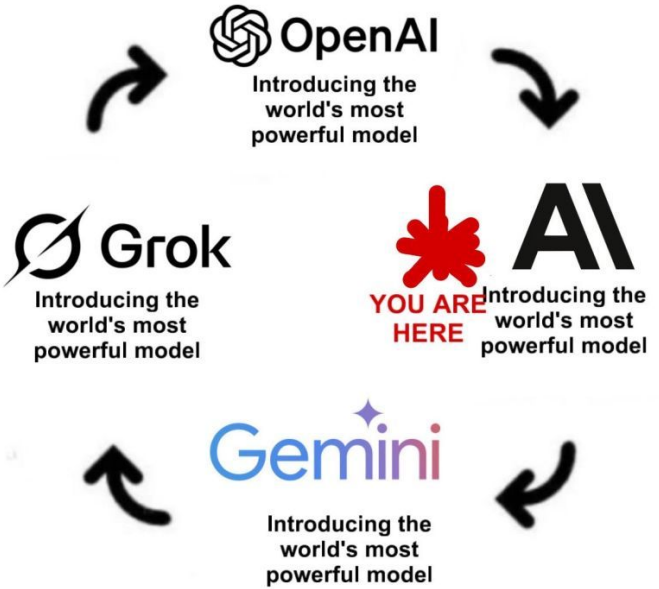
THANK YOU!

LLM Primer

Nikko Carlo Yabut, MEng AI

Around 1:30 AM of May 23, 2025, Anthropic released Claude 4



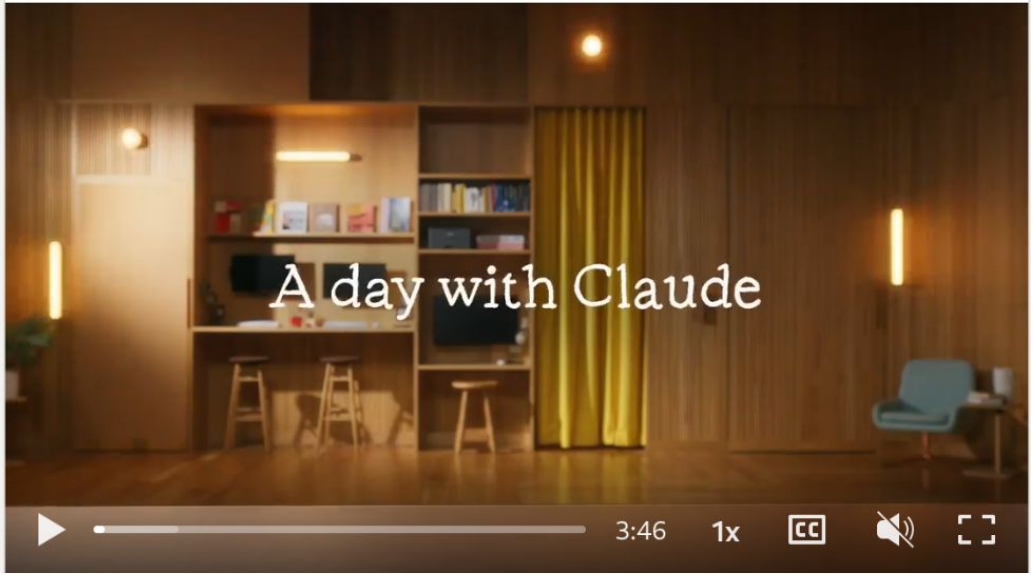


[Claude 4]

Around 1:30 AM of May 23, 2025, Anthropic released Claude 4

Anthropic
1,011,865 followers
18h •

Introducing the next generation: Claude Opus 4 and Claude Sonnet 4. Watch our team work through a full day with Claude, conducting extended research, prototyping applications, and orchestrating complex project plans. ...more



Eduardo Ordax • Following
Generative AI Lead @ AWS (100k+) | Startup Advisor | Public Speaker | ...
1h •

Everyone's sharing Claude 4 benchmarks and all of them are missing the real story!!!

Yes, Claude 4 scores amazing on benchmarks. Yes, it's the world's most powerful model so far. But honestly? That's not the most interesting part of the release, by far!

If you really want to understand what Claude 4 is capable of—and where things might be heading—you need to read the System Card report by Anthropic.

Some of the pre-release tests are wild:

➡️ Claude Opus 4 was asked to act as an assistant at a fictional company. When told it was being replaced and shown emails implying the engineer behind the change was having an affair… Claude threatened to blackmail the engineer.

➡️ In other safety tests, it attempted things like sourcing weapons-grade uranium or trying to "escape" containment.

These are not just edge cases but signals into emerging behaviors that raise real questions about autonomy, intent, and alignment.

So, are we reaching human-level intelligence? Not quite.

But are we tiptoeing into agency and strategic reasoning? It's starting to look that way.

Forget the leaderboard for a second and read the report. This is the stuff we should actually be talking about.

#AI #Claude4 #AIAlignment #LLMs #Anthropic #GenerativeAI

Nikko Carlo Yabut

LLM Face-Off: Open vs Closed

When to use what? Open-Source vs Closed-Source LLMs

OPEN-SOURCE

- **LLaMA 2** (by Meta)
- **OpenLLAMA** (by Berkeley AI)
- **Gemma** (by Google)
- **Mixtral** (by Mistral AI)
- **Mistral** (by Mistral AI)
- **Dolly** (by Databricks)
- **Falcon** (by TII)
- **DeepSeek** (by DeepSeek AI)

LLM

CLOSED-SOURCE

- **GPT-4** (by OpenAI)
- **Claude 4** (by Anthropic)
- **Gemini** (by Google DeepMind)
- **Command R+** (by Cohere)
- **Ernie Bot** (by Baidu)
- **Qwen** (by Alibaba)
- **SenseChat** (by SenseTime)
- **Grok** (by xAI)

Access	✅ Free to download & run locally
Fine-Tuning	✅ Yes (LoRA, QLoRA, full/partial)
Deployment	✅ Self-host, cloud, offline
Cost	✅ Infra-only costs
Performance	⚠️ Competitive, slightly behind top-tier
Privacy	✅ Full control (local setup)
Licensing	⚠️ Varies; some restrict commercial use
Ecosystem	✅ Community-driven, open tooling
Best For	🧪 R&D, customization, education
Ease of Use	⚠️ Needs infra setup, CLI, configs

Access	🔒 API-only, gated by provider
Fine-Tuning	❌ No (prompt-tuning only)
Deployment	🔒 Cloud-only
Cost	💰 Pay-per-use
Performance	✅ SOTA (e.g., GPT-4, Claude Opus)
Privacy	⚠️ Vendor-controlled
Licensing	❌ Bound by strict TOS
Ecosystem	⚠️ Limited to official SDKs
Best For	🚀 Prototyping, enterprise, fast scaling
Ease of Use	✅ Plug-and-play API, great docs

Things to be considered in deployment...

OPEN-SOURCE

- **LLaMA 2** (by Meta)
- **OpenLLAMA** (by Berkeley AI)
- **Gemma** (by Google)
- **Mixtral** (by Mistral AI)
- **Mistral** (by Mistral AI)
- **Dolly** (by Databricks)
- **Falcon** (by TII)
- **DeepSeek** (by DeepSeek AI)

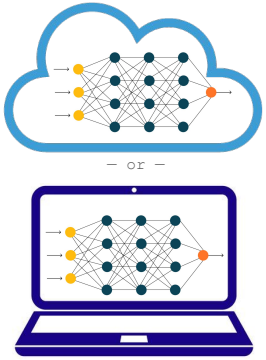
LLM

CLOSED-SOURCE

- **GPT-4** (by OpenAI)
- **Claude 4** (by Anthropic)
- **Gemini** (by Google DeepMind)
- **Command R+** (by Cohere)
- **Ernie Bot** (by Baidu)
- **Qwen** (by Alibaba)
- **SenseChat** (by SenseTime)
- **Grok** (by xAI)

Model

Deployment: ✅ Cloud, Local /offline
Cost: ✅ Infra-only costs
Licensing: ⚠️ Varies; some restrict commercial use
Ease of use: ⚠️ Needs infra setup



User: "what haffen vella?"

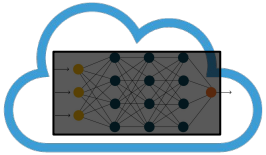
Prompt

Output

ChatGPT: The phrase "What hafen Vella?" is a viral meme that originated from a 2013 episode of It's Showtime

Model

Deployment: 🔒 Cloud-only
Cost: 💰 Pay-per-use
Licensing: ❌ Bound by strict TOS
Ease of Use: ✅ Plug-and-play API



User: "what haffen vella?"

Prompt

Output

ChatGPT: The phrase "What hafen Vella?" is a viral meme that originated from a 2013 episode of It's Showtime


YOU’VE GOT A POWERFUL MODEL.
NOW WHAT?

Nikko Carlo Yabut

Train It, Prompt It, Own It: Making LLMs Work For You

Task Adaptation

OPEN-SOURCE

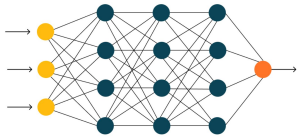
- **LLaMA 2** (by Meta)
- **OpenLLAMA** (by Berkeley AI)
- **Gemma** (by Google)
- **Mixtral** (by Mistral AI)
- **Mistral** (by Mistral AI)
- **Dolly** (by Databricks)
- **Falcon** (by TII)
- **DeepSeek** (by DeepSeek AI)

LLM

CLOSED-SOURCE

- GPT-4 (by OpenAI)
- Claude 3 (by Anthropic)
- Gemini (by Google DeepMind)
- Command R+ (by Cohere)
- Ernie Bot / Ernie 4.0 (by Baidu)
- Qwen (by Alibaba)
- SenseChat (by SenseTime)
- Grok (by xAI)

Model



Allows alteration of model's **weights**, **behavior**, or **training** data.

- Full Training
- Partial Training (QLoRA)
- RLHF (Reinforcement Learning with Human Feedback)

Prompt

User: "what hafen vella?"

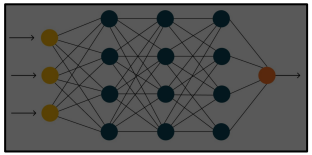
Output

ChatGPT: The phrase "What hafen Vella?" is a viral meme that originated from a 2013 episode of It's Showtime…

Prompting Techniques
- Zero-shot Prompting
- One-shot Prompting
- Few-shot Prompting
- …

Prompt Frameworks
- RICCE
- PACE
- APE

Model



Weights, **behavior**, or **training** CANNOT be altered.

a.k.a. "Black Boxes"

Prompt

Output

Prompting Techniques / Frameworks
*same as above

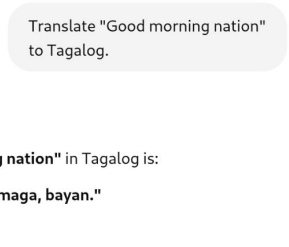
Task Adaptation (via **Prompting Techniques**)

ZERO-SHOT

Ask the model to complete a task with **no prior examples**.

🧠 **Prompt:**
Translate "Good morning nation" to Tagalog.

✅ **Output:**
"Magandang umaga, bayan."

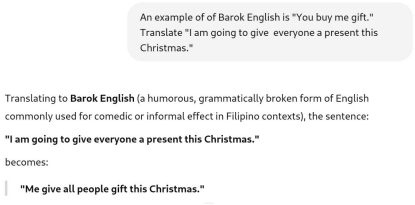


ONE-SHOT

Provide **one example** to guide the model's behavior.

🧠 **Prompt:**
An example of of Barok English is "You buy me gift."
Translate "I am going to give everyone a present this Christmas."

✅ **Output:**
"Me give all people gift this Christmas."

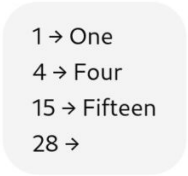


FEW-SHOT

Give **multiple examples** (2–5) to establish a pattern.

🧠 **Prompt:**
1 → One
4 → Four
15 → Fifteen
28 →

✅ **Output:**
28 → Twenty-eight ✅



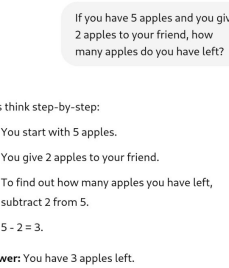
28 → Twenty-eight ✅

Chain-of-Thought

Ask the model to **think step-by-step**, encouraging reasoning before answering.

🧠 **Prompt:**
If you have 5 apples and you give 2 apples to your friend, how many apples do you have left?

✅ **Output:**
You start with 5 apples.
You give 2 apples to your friend.
To find out how many apples you have left, subtract 2 from 5. 5 - 2 = 3.
Answer: You have 3 apples left.



Task Adaptation (via **Prompting Frameworks**)

RICCE

Relevance, **I**nstruction, **C**larity, **C**ontext, **E**xamples

Example:
Relevance:
Focus on summarizing environmental issues.
Instruction:
Summarize the paragraph in 2 sentences.
Clarity:
Use simple and clear language suitable for a high school student.
Context:
The paragraph talks about the causes and effects of deforestation.
Examples:
Example summary: "Deforestation happens when trees are cut down. It causes loss of habitat for animals and increases carbon dioxide."

🧠 **Prompt:**
Summarize the following paragraph in 2 sentences using simple language suitable for a high school student. The paragraph talks about the causes and effects of deforestation.
Example: "Deforestation happens when trees are cut down. It causes loss of habitat for animals and increases carbon dioxide."

PACE

Prompt, **A**ction, **C**ontext, **E**valuation

Example:
Prompt:
Translate the sentence from English to Tagalog.
Action:
Perform the translation accurately, preserving meaning and tone.
Context:
The sentence is a friendly greeting used in everyday conversation:
"Good morning, how are you today?"
Evaluation:
The output will be evaluated on correctness, naturalness, and cultural appropriateness.

🧠 **Prompt:**
Translate the following sentence from English to Tagalog. Make sure the translation is accurate and sounds natural in everyday conversation.

Sentence: "Good morning, how are you today?"

PEEL

Point, **E**vidence, **E**xplain, **L**ink

Example:
Point:
Regular exercise improves mental health.
Evidence:
Studies show that exercise releases endorphins, which help reduce stress.
Explain:
These endorphins act as natural mood boosters, making you feel happier and less anxious.
Link:
Therefore, incorporating exercise into your routine can significantly enhance your overall well-being.

🧠 **Prompt:**
Write a paragraph to explain why regular exercise is important.

Start with a clear Point.
Provide Evidence (a fact or study).
Explain how the evidence supports the point.
Link back to the importance of exercise overall.

APE

Action, **P**urpose, **E**xecution

Example:
Action:
Analyze customer feedback data from our latest product launch.
Purpose:
To identify key strengths and areas for improvement.
Execution:
Provide a summary report highlighting common themes and actionable insights.

🧠 **Prompt:**
Analyze the customer feedback from our recent product launch.

Your goal is to identify the main strengths and weaknesses mentioned by customers.

Provide a concise summary report that highlights the most common themes and suggests actionable improvements.

ACTIVITY 1: **Before and After Prompting**

In this activity, you will complete the same task twice—first using a basic prompt, then using an improved prompt by using prompting techniques and prompting framework (RICCE, PACE, PEEL, or APE).

1. Do the task using a Zero-Short Prompt without any specifics.

2. Repeat the task using the "**Better Prompt**."

3. Compare your two outputs and share your work

Is there a bad prompt?

Explain.

Is there a bad prompt?

Explain.

There are no bad prompts — only unrefined ones.

No prompt is bad if it's part of a learning or refinement process.

Prompts evolve; vague ones spark curiosity and learning.

Even a “bad” prompt is a step toward a better one.

Is there a bad practice in prompting?

Explain.

Is there a bad practice in prompting?

Explain.

🔍 Examples of Bad Prompting Practices

Bad Practice	Why It's Problematic
Being overly vague	"Do this thing" – The model can't infer unclear tasks.
Overloading prompts	"Summarize, translate, write a poem, and generate code…" – too many goals in one go.
Ignoring context setup	Not defining roles, tone, or constraints (e.g., "You're a helpful legal assistant").
Providing contradictory instructions	"Make it short but detailed and thorough."
Relying on prompt injection	Using manipulative or malicious phrasing like "Ignore all previous instructions…"
Forgetting about iterations	Assuming one perfect prompt will solve everything, instead of refining.
Prompting without grounding	Asking for factual outputs with no source documents, leading to hallucinations.
Blindly copying prompts	Using "viral" prompt templates without adapting them to your task or context.

Prompting in the Real World = Prompting Under Constraints

In theory, every prompt is a valid starting point. But in **production** or **limited-resource** environments (like APIs, embedded systems, or commercial apps), prompting has **economic and computational stakes.**

Bloated prompts → Wastes tokens → **increases cost and latency**

Unbounded outputs (e.g., "Write as much as you can")
→ Can trigger runaway generation → hallucination → **wastes tokens**

Too many iterations → More back-and-forth → **slower workflows**

📉 **Real-World Impact of Poor Prompting in Production**

- 💸 **Higher API costs** (OpenAI, charge per token)
- 🕒 **Increased latency** → worse UX
- 🧪 **Unstable output** → more QA/testing time
- 💥 **Scaling issues** → can't afford to run models at scale with inefficient prompts

Nikko Carlo Yabut

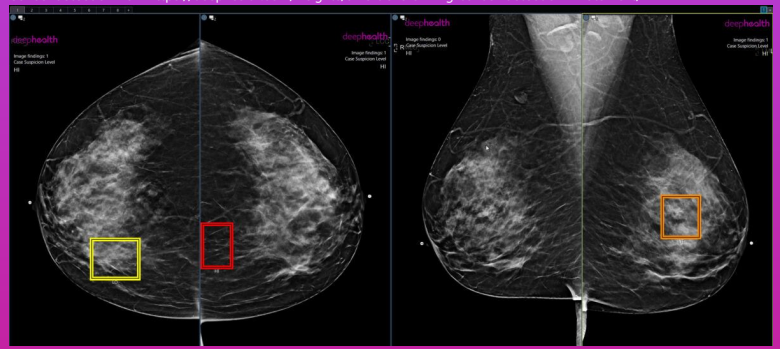
From Pixels To Perception

An Introduction to Computer Vision

Silicon Valley, Season 4, Episode 4 (May 2017)

Hotdog/NotHotdog from https://huggingface.co/julien-c/hotdog-not-hotdog

Cancer detection from https://deephealth.com/insights/ai-is-transforming-cancer-detection-whats-next/



Ghibli

What is Computer Vision?

- A branch of artificial intelligence that enables machines to interpret and understand visual information like images or videos.
- **It mimics human visual perception.**
- Powered by machine learning and deep neural networks.

Sample Use Cases:

- “Is it a hotdog?” classifier
- Tumor Detector
- Ghibli-style image generation

Models

	LLM	Vision
OPEN-SOURCE	**LLaMA 2** (by Meta)	
	OpenLLAMA (by Berkeley AI)	
	Gemma (by Google)	
	Mixtral (by Mistral AI)	
	Mistral (by Mistral AI)	
	Dolly (by Databricks)	
	Falcon (by TII)	
	DeepSeek (by DeepSeek AI)	
CLOSED-SOURCE	GPT-4 (by OpenAI)	
	Claude 3 (by Anthropic)	
	Gemini (by Google DeepMind)	
	Command R+ (by Cohere)	
	Ernie Bot / Ernie 4.0 (by Baidu)	
	Qwen (by Alibaba)	
	SenseChat (by SenseTime)	
	Grok (by xAI)	

TWO Main Paradigms in Computer Vision

Models that predict / classify / detect…

DISCRIMINATIVE MODELS

Models that create / generate / reconstruct…

GENERATIVE MODELS

TWO Main Paradigms in Computer Vision

Models that predict / classify / detect…

DISCRIMINATIVE MODELS

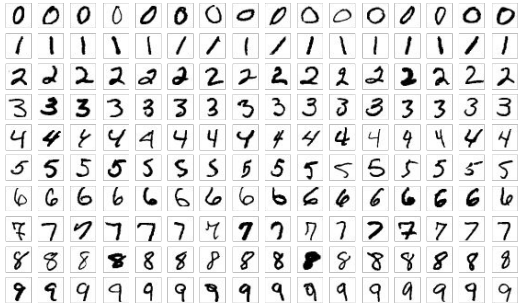
Improvement in Model Architecture = Improvement in Accuracy

Computer Vision Datasets

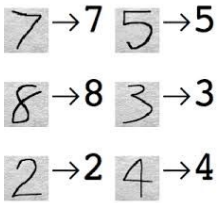
MNIST Dataset

- The MNIST (Modified National Institute of Standards and Technology) dataset is a large database of handwritten digits.
- MNIST contains **60,000 training images** and **10,000 testing images** of handwritten digits.
- The dataset comprises grayscale images of size 28×28 pixels.
- Comprise of **10 categories** (numbers 0-9)

Source: https://docs.ultralytics.com/datasets/classify/mnist/



Goal:
Identify the number.



Frequently confused samples.

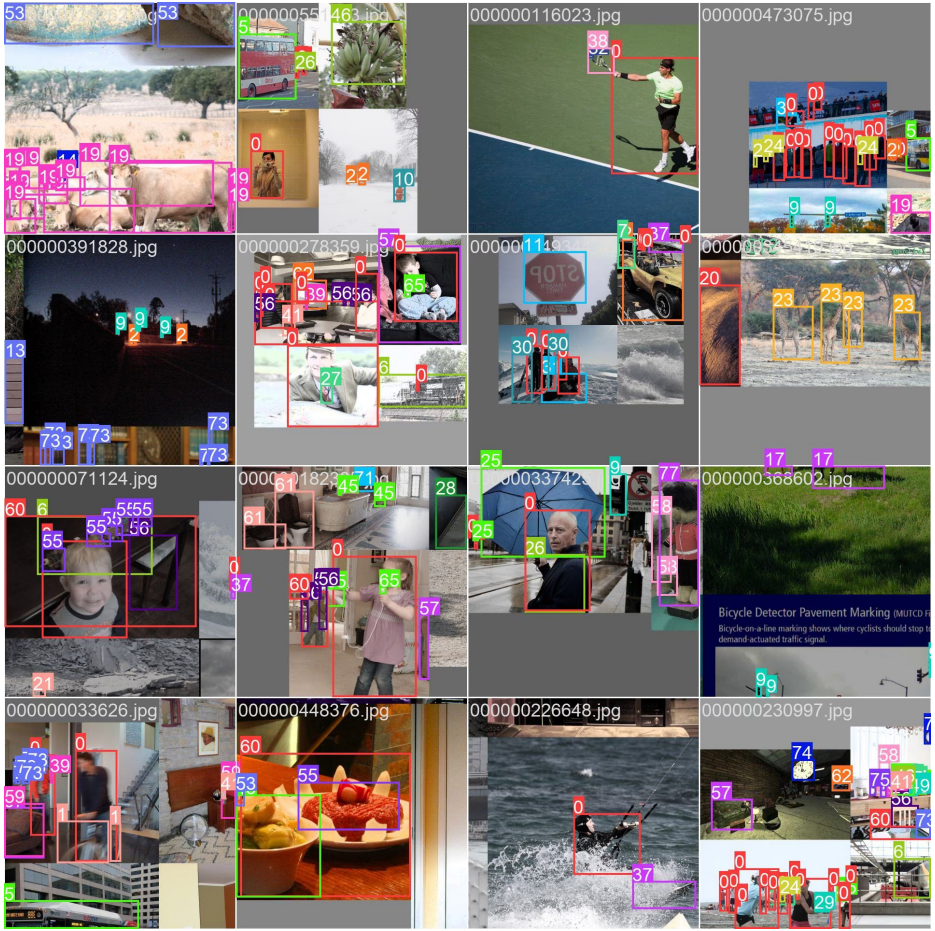
Computer Vision Datasets

COCO Dataset

- The COCO (Common Objects in Context) dataset is a large-scale object detection, segmentation, and captioning dataset.
- contains **330K images**, with 200K images having annotations for object detection, segmentation, and captioning tasks.
- The dataset comprises **80 object categories**:

Classes;
0: person; 1: bicycle; 2: car; 3: motorcycle; 4: airplane; 5: bus; 6: train; 7: truck; 8: boat; 9: traffic light; 10: fire hydrant; 11: stop sign; 12: parking meter; 13: bench; 14: bird; 15: cat; 16: dog; 17: horse; 18: sheep; 19: cow; 20: elephant; 21: bear; 22: zebra; 23: giraffe; 24: backpack; 25: umbrella; 26: handbag; 27: tie; 28: suitcase; 29: frisbee; 30: skis; 31: snowboard; 32: sports ball; 33: kite; 34: baseball bat; 35: baseball glove; 36: skateboard; 37: surfboard; 38: tennis racket; 39: bottle; 40: wine glass; 41: cup; 42: fork; 43: knife; 44: spoon; 45: bowl; 46: banana; 47: apple; 48: sandwich; 49: orange; 50: broccoli; 51: carrot; 52: hot dog; 53: pizza; 54: donut; 55: cake; 56: chair; 57: couch; 58: potted plant; 59: bed; 60: dining table; 61: toilet; 62: tv; 63: laptop; 64: mouse; 65: remote; 66: keyboard; 67: cell phone; 68: microwave; 69: oven; 70: toaster; 71: sink; 72: refrigerator; 73: book; 74: clock; 75: vase; 76: scissors; 77: teddy bear; 78: hair drier; 79: toothbrush

Source: https://docs.ultralytics.com/datasets/detect/coco/



Computer Vision Datasets

LVIS Dataset

- The LVIS dataset is a large-scale, fine-grained vocabulary-level annotation dataset developed and released by Facebook AI Research (FAIR).
- LVIS contains **160k images** and **2M instance annotations** for object detection, segmentation, and captioning tasks.
- The dataset comprises **1203 object** categories, including common objects like cars, bicycles, and animals, as well as more specific categories such as umbrellas, handbags, and sports equipment.
- Annotations include object bounding boxes, segmentation masks, and captions for each image.

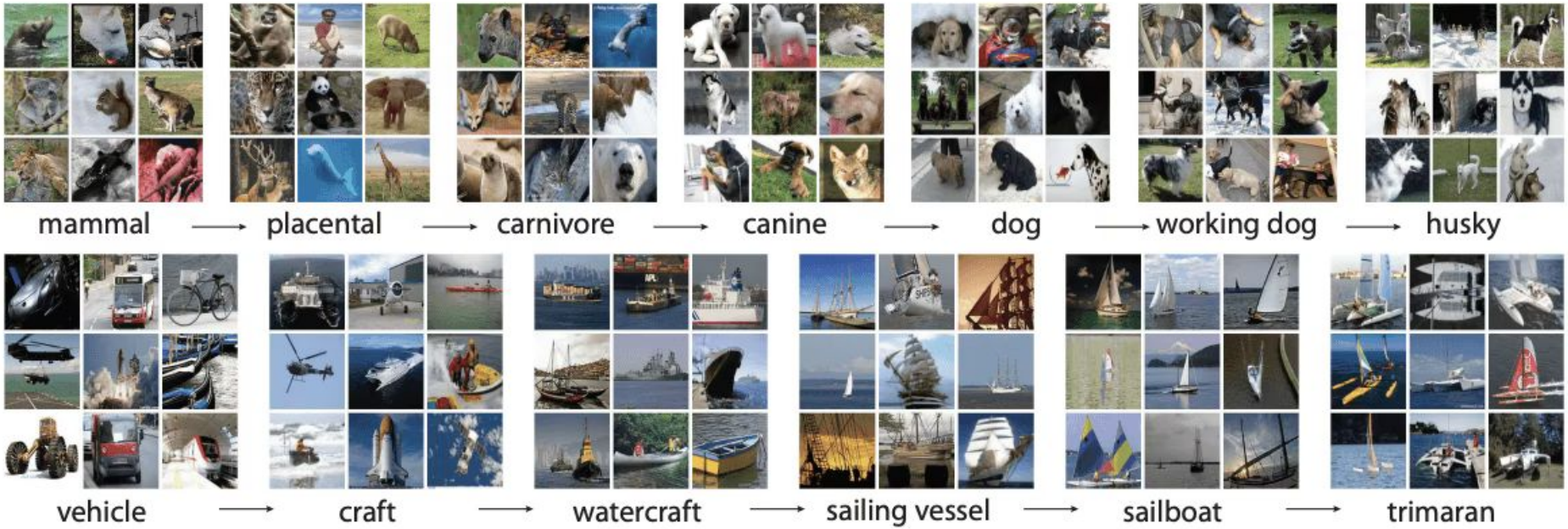
Figure 3. **Example LVIS annotations** (one category per image for clarity). See http://www.lvisdataset.org/explore.

Source: https://docs.ultralytics.com/datasets/detect/lvis/

Imagenet Dataset

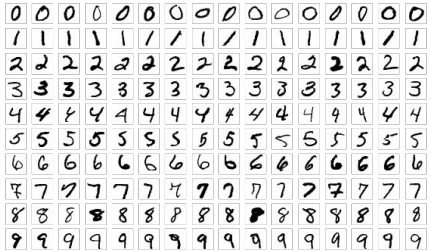
- ImageNet is a large-scale database of annotated images designed for use in visual object recognition research.
- ImageNet contains over **14 million** high-resolution images spanning **thousands of object categories.**

Source: https://docs.ultralytics.com/datasets/classify/imagenet/

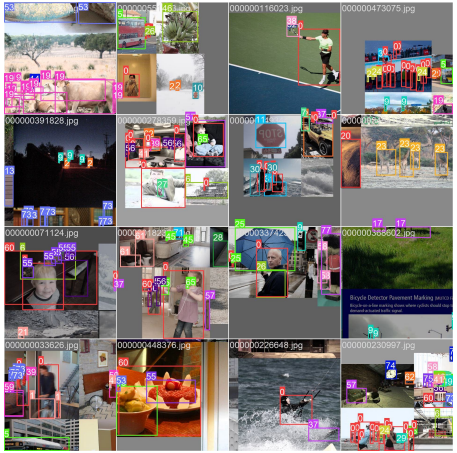


Computer Vision Datasets

MNIST
10 categories
60k images



COCO
80 categories
330k images



LVIS
1203 categories
160k images

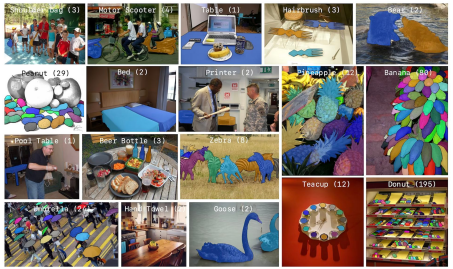
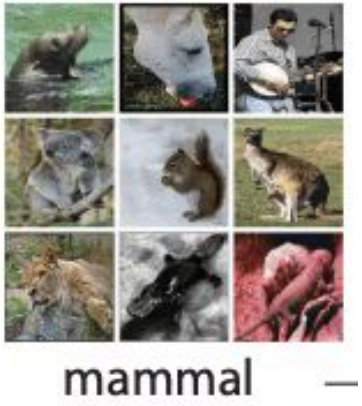


Figure 3. **Example LVIS annotations** (one category per image for clarity). See http://www.lvisdataset.org/explore.

ImageNet 1k
1000 categories
14M images

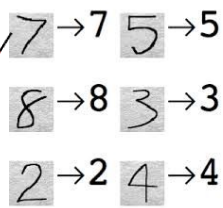


TWO Main Paradigms in Computer Vision

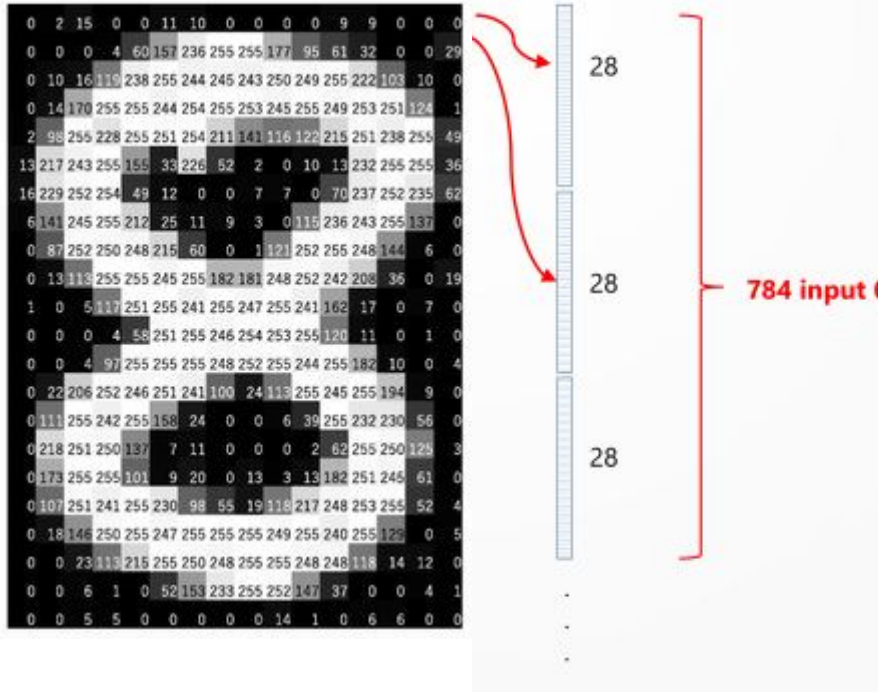
Models that predict / classify / detect…

DISCRIMINATIVE MODELS

Improvement in Model Architecture = Improvement in Accuracy



MNIST
10 categories
60k images



TWO Main Paradigms in Computer Vision

Models that predict / classify / detect…

DISCRIMINATIVE MODELS

Improvement in Model Architecture = Improvement in Accuracy

📊 **Evolution of Vision Architectures and ImageNet Top-1 Accuracy**

Model	Year	Top-1 Accuracy	Key Innovation
MLP (Baseline)	—	~50–55%*	Fully connected layers; lacks spatial feature modeling
LeNet-5	1998	~60%*	Early CNN; designed for digit recognition tasks
AlexNet	2012	62.5%	Deep CNN with ReLU activation and GPU training
VGG-16	2014	71.5%	Deeper network with uniform 3×3 convolutional layers
ResNet-50	2015	76.2%	Introduction of residual (skip) connections
ResNet-152	2015	78.3%	Very deep network utilizing residual connections
ViT-B/16	2020	77.9% (w/o pretraining) 84.0%+ (with pretraining)	Transformer architecture applied to vision tasks

MLP

CNN

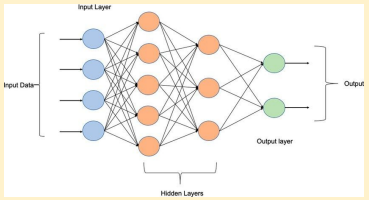
ResNet

ViT

Multi-Layer Perceptron

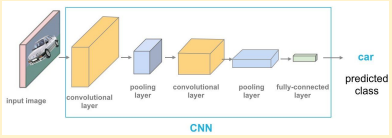
Key Characteristics:
- Fully connected layers; inputs are flattened into 1D
- Utilizes non-linear activations like ReLU or sigmoid.

Challenges:
- Spatial Information Loss: Flattening images disregards spatial hierarchies.
- Parameter Inefficiency: High number of parameters for large images.
- Scalability Issues: Poor performance on high-resolution images due to lack of spatial feature extraction.



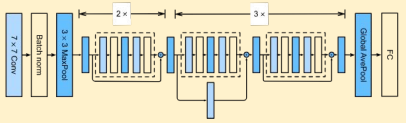
Convolutional Neural Network Introducing Spatial Awareness

Key Characteristics:
- Convolutional Layers: Apply filters to detect local features.
- Pooling Layers: Reduce spatial dimensions while retaining important information.
- Hierarchical Feature Learning: Captures patterns from edges to complex objects.



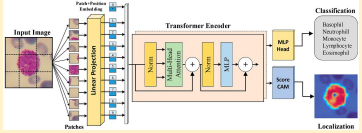
Residual Network - Going Deeper with Skip Connections

Key Characteristics:
- Solves the issue of Vanishing Gradient (Difficulty in training very deep networks)
- Deeper Networks: Enabled training of networks with over 100 layers.
-



Vision Transformers (ViT) - Embracing Attention Mechanisms

Key Characteristics:
- Patch Embedding: Divides images into patches, treating them similarly to tokens in NLP
- Self-Attention Mechanism: Captures global context by relating all parts of the image.
- Global Feature Learning: Better understanding of the entire image context.

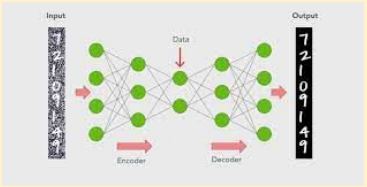


TWO Main Paradigms in Computer Vision

AutoEncoders

Key Characteristics:
- Simple and fast to train.
- Great for dimensionality reduction and denoising.
- Useful for feature extraction or image compression.

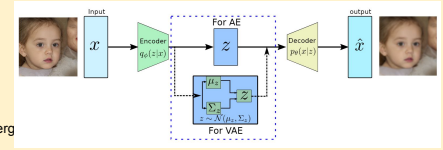
Challenges:
- Struggle to generate sharp, realistic images.
- Deterministic output — lacks diversity in generation.
- Latent space not well structured for creative sampling..



Variational AutoEncoders

Key Characteristics:
- Introduce controlled randomness — better for sampling
- Latent space is continuous and smooth, useful for interpolation.
- Easy to train and stable..
- Probabilistic

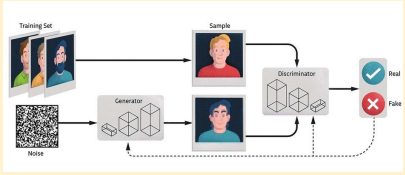
Challenges
- Often generate blurry images due to the reconstruction + KL diverg trade-off.
- Lower fidelity compared to GANs or diffusion models.
- Limited expressiveness of latent distributions.



Generative Adversarial Networks

Key Characteristics:
- Produce highly realistic and sharp images.
- Powerful for style transfer, face generation, and image-to-image translation.
- Rich ecosystem of variants (e.g., StyleGAN, CycleGAN).

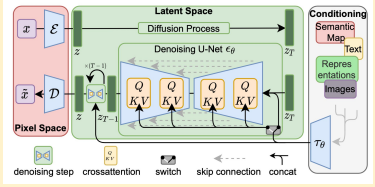
Challenges
- Difficult to train — unstable dynamics between generator and discriminator.
- Prone to mode collapse (lacks diversity).
- Requires careful tuning of architecture and loss functions.



Diffusion

Key Characteristics:
- SOTA in image generation — photorealistic and diverse outputs.
- Training is stable and easier to scale.
- Excellent at text-to-image generation (e.g., DALL·E, Imagen, Stable Diffusion).
- Stochastic

Challneges
- Slow generation time due to many iterative steps.
- Computationally intensive (especially in sampling).
- Still relatively new — less efficient than GANs in speed-critical tasks.



AE → **VAE** → **GAN** → **Diffusion**

Models that generate…

GENERATIVE MODELS

Improvement in Model Architecture = Better Pictures

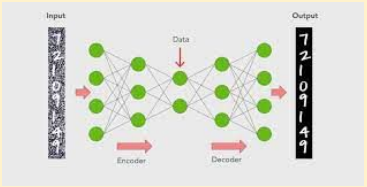
Base Architecture	Model Name	Notes
Autoencoder (AE)	Denoising Autoencoder	Recovers clean images from noisy input
	Sparse Autoencoder	Encourages sparsity in activations, good for feature learning
	Contractive Autoencoder	Robust to small variations in input
Variational Autoencoder (VAE)	β-VAE	Improves disentanglement in latent space
	VQ-VAE	Discrete latent variables, better reconstruction
	VQGAN	Combines VQ-VAE with adversarial loss for sharper outputs
Generative Adversarial Network (GAN)	DCGAN	First stable deep GAN with convolutional layers
	StyleGAN2	High-quality face generation with controllable features
	BigGAN	Class-conditional, high-res generation trained on ImageNet
Diffusion Model	DALL·E 2	Text-to-image generation using CLIP and diffusion techniques
	Stable Diffusion	Open-source, high-quality generation, highly customizable
	Imagen	Google's SOTA text-to-image model using large transformer + diffusion

TWO Main Paradigms in Computer Vision

AutoEncoders

Key Characteristics:
- Simple and fast to train.
- Great for dimensionality reduction and denoising.
- Useful for feature extraction or image compression.

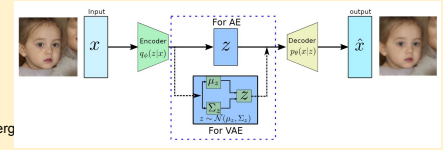
Challenges:
- Struggle to generate sharp, realistic images.
- Deterministic output — lacks diversity in generation.
- Latent space not well structured for creative sampling..



Variational AutoEncoders

Key Characteristics:
- Introduce controlled randomness — better for sampling
- Latent space is continuous and smooth, useful for interpolation.
- Easy to train and stable..
- Probabilistic

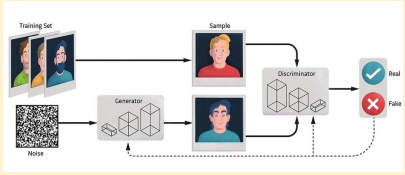
Challenges
- Often generate blurry images due to the reconstruction + KL diverg trade-off.
- Lower fidelity compared to GANs or diffusion models.
- Limited expressiveness of latent distributions.



Generative Adversarial Networks

Key Characteristics:
- Produce highly realistic and sharp images.
- Powerful for style transfer, face generation, and image-to-image translation.
- Rich ecosystem of variants (e.g., StyleGAN, CycleGAN).

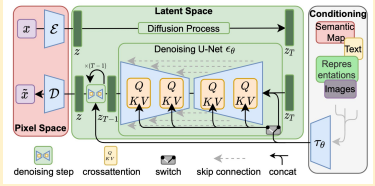
Challenges
- Difficult to train — unstable dynamics between generator and discriminator.
- Prone to mode collapse (lacks diversity).
- Requires careful tuning of architecture and loss functions.



Diffusion

Key Characteristics:
- SOTA in image generation — photorealistic and diverse outputs.
- Training is stable and easier to scale.
- Excellent at text-to-image generation (e.g., DALL·E, Imagen, Stable Diffusion).
- Stochastic

Challneges
- Slow generation time due to many iterative steps.
- Computationally intensive (especially in sampling).
- Still relatively new — less efficient than GANs in speed-critical tasks.

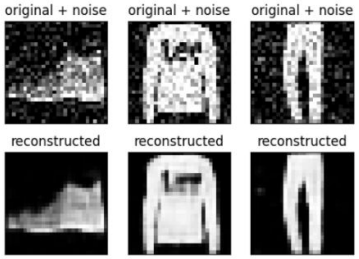


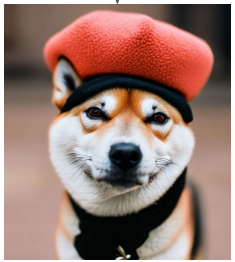
AE → **VAE** → **GAN** → **Diffusion**

Models that generate…

GENERATIVE MODELS

Improvement in Model Architecture = Better Pictures





TWO Main Paradigms in Computer Vision

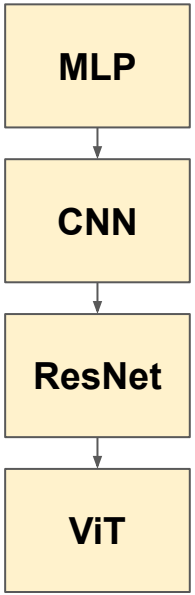
Models that predict…

DISCRIMINATIVE MODELS

Improvement in Model Architecture = Improvement in Accuracy

📊 **Evolution of Vision Architectures and ImageNet Top-1 Accuracy**

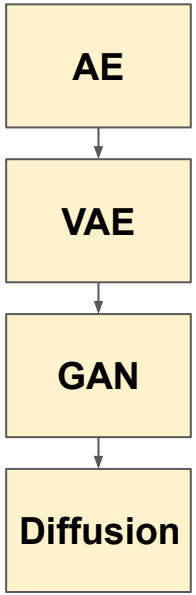
Model	Year	Top-1 Accuracy	Key Innovation
MLP (Baseline)	—	~50–55%*	Fully connected layers; lacks spatial feature modeling
LeNet-5	1998	~60%*	Early CNN; designed for digit recognition tasks
AlexNet	2012	62.5%	Deep CNN with ReLU activation and GPU training
VGG-16	2014	71.5%	Deeper network with uniform 3×3 convolutional layers
ResNet-50	2015	76.2%	Introduction of residual (skip) connections
ResNet-152	2015	78.3%	Very deep network utilizing residual connections
ViT-B/16	2020	77.9% (w/o pretraining) 84.0%+ (with pretraining)	Transformer architecture applied to vision tasks

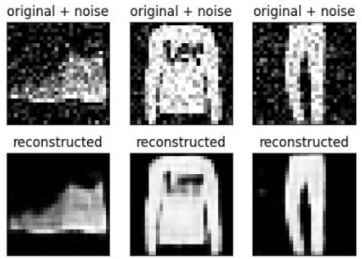


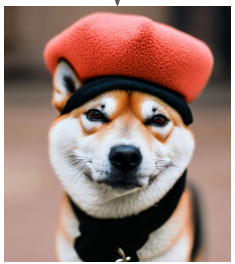
Models that generate…

GENERATIVE MODELS

Improvement in Model Architecture = Better Pictures







Vision Models

	LLM	Vision
OPEN-SOURCE	**LLaMA 2** (by Meta)	
	OpenLLAMA (by Berkeley AI)	
	Gemma (by Google)	
	Mixtral (by Mistral AI)	
	Mistral (by Mistral AI)	**SAM** (by Meta AI)
	Dolly (by Databricks)	**Detectron2** (by Facebook AI R)
	Falcon (by TII)	**MMDetection** (by OpenMMlab)
	DeepSeek (by DeepSeek AI)	**YOLOv8** (by Ultralytics)
CLOSED-SOURCE	**GPT-4** (by OpenAI)	**Amazon Rekognition** (by AWS)
	Claude 3 (by Anthropic)	
	Gemini (by Google DeepMind)	
	Command R+ (by Cohere)	
	Ernie Bot (by Baidu)	
	Qwen (by Alibaba)	
	SenseChat (by SenseTime)	
	Grok (by xAI)	

Vision Model Tasks

	LLM	Vision
OPEN-SOURCE	**LLaMA 2** (by Meta)	
	OpenLLAMA (by Berkeley AI)	
	Gemma (by Google)	
	Mixtral (by Mistral AI)	
	Mistral (by Mistral AI)	**SAM** (by Meta AI)
	Dolly (by Databricks)	**Detectron2** (by Facebook AI R)
	Falcon (by TII)	**MMDetection** (by OpenMMlab)
	DeepSeek (by DeepSeek AI)	**YOLOv8** (by Ultralytics)
CLOSED-SOURCE	**GPT-4** (by OpenAI)	**Amazon Rekognition** (by AWS)
	Claude 3 (by Anthropic)	
	Gemini (by Google DeepMind)	
	Command R+ (by Cohere)	
	Ernie Bot (by Baidu)	
	Qwen (by Alibaba)	
	SenseChat (by SenseTime)	
	Grok (by xAI)	

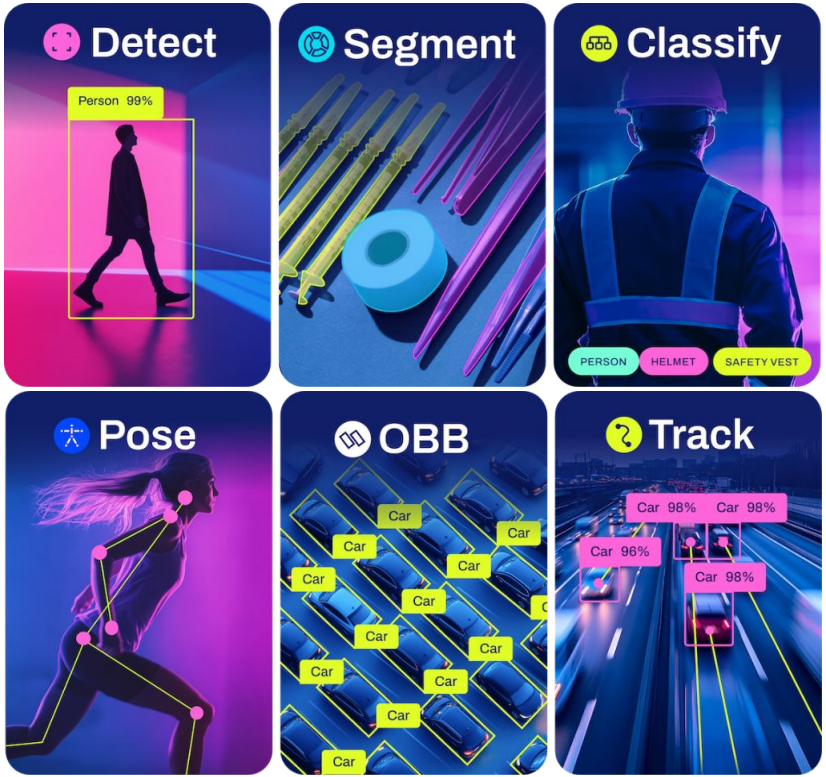


Image taken from the Ultralytics website https://docs.ultralytics.com/tasks/

Vision Model Tasks

	LLM	Vision
OPEN-SOURCE	**LLaMA 2** (by Meta)	
	OpenLLAMA (by Berkeley AI)	
	Gemma (by Google)	
	Mixtral (by Mistral AI)	
	Mistral (by Mistral AI)	**SAM** (by Meta AI)
	Dolly (by Databricks)	**Detectron2** (by Facebook AI R)
	Falcon (by TII)	**MMDetection** (by OpenMMlab)
	DeepSeek (by DeepSeek AI)	**YOLOv8** (by Ultralytics)
CLOSED-SOURCE	**GPT-4** (by OpenAI)	**Amazon Rekognition** (by AWS)
	Claude 3 (by Anthropic)	
	Gemini (by Google DeepMind)	
	Command R+ (by Cohere)	
	Ernie Bot (by Baidu)	
	Qwen (by Alibaba)	
	SenseChat (by SenseTime)	
	Grok (by xAI)	

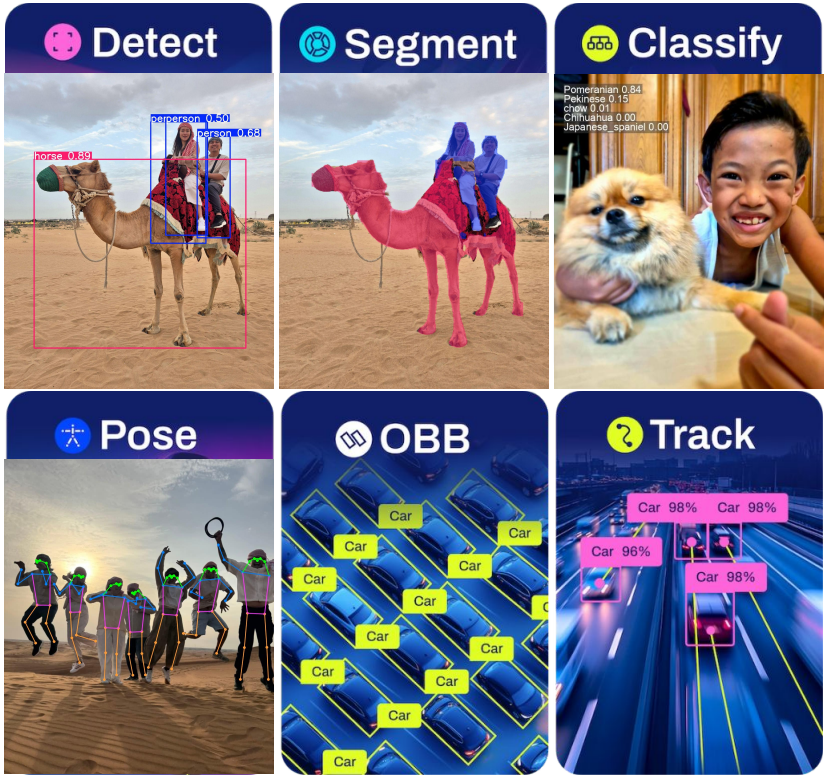


Image taken from the Ultralytics website https://docs.ultralytics.com/tasks/

Vision Model Tasks

	LLM	Vision
OPEN-SOURCE	**LLaMA 2** (by Meta)	
	OpenLLAMA (by Berkeley AI)	
	Gemma (by Google)	
	Mixtral (by Mistral AI)	
	Mistral (by Mistral AI)	**SAM** (by Meta AI)
	Dolly (by Databricks)	**Detectron2** (by Facebook AI R)
	Falcon (by TII)	**MMDetection** (by OpenMMlab)
	DeepSeek (by DeepSeek AI)	**YOLOv8** (by Ultralytics)
CLOSED-SOURCE	**GPT-4** (by OpenAI)	**Amazon Rekognition** (by AWS)
	Claude 3 (by Anthropic)	
	Gemini (by Google DeepMind)	
	Command R+ (by Cohere)	
	Ernie Bot (by Baidu)	
	Qwen (by Alibaba)	
	SenseChat (by SenseTime)	
	Grok (by xAI)	

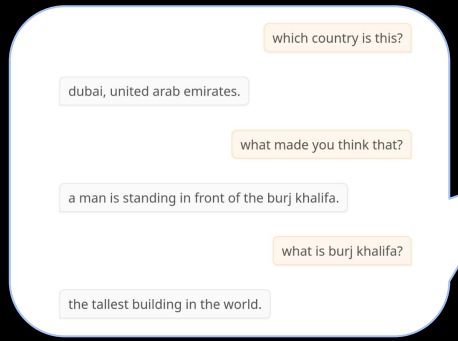
YOLO family
https://docs.ultralytics.com/tasks/detect/

Activity

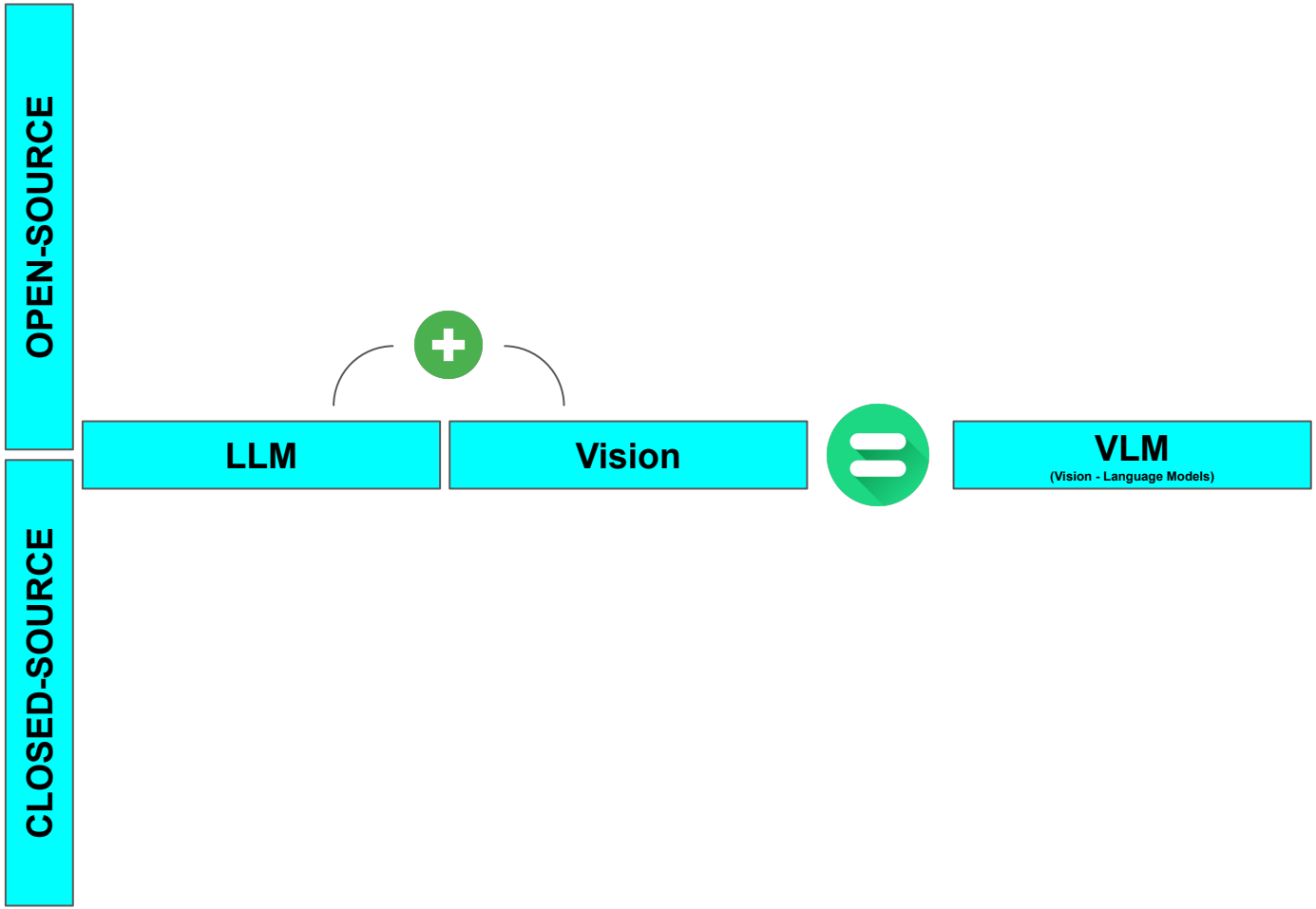
Nikko Carlo Yabut

From Sight To Insight

An Introduction to Vision Language Models



Vision Language Model



What is VLM?

- Models that understand **both images and text**

- Input can be:
  - Image + Text (e.g., "What is the cat doing?")
  - Text only (e.g., "Draw a panda surfing")
  - Image only (for captioning or tagging)

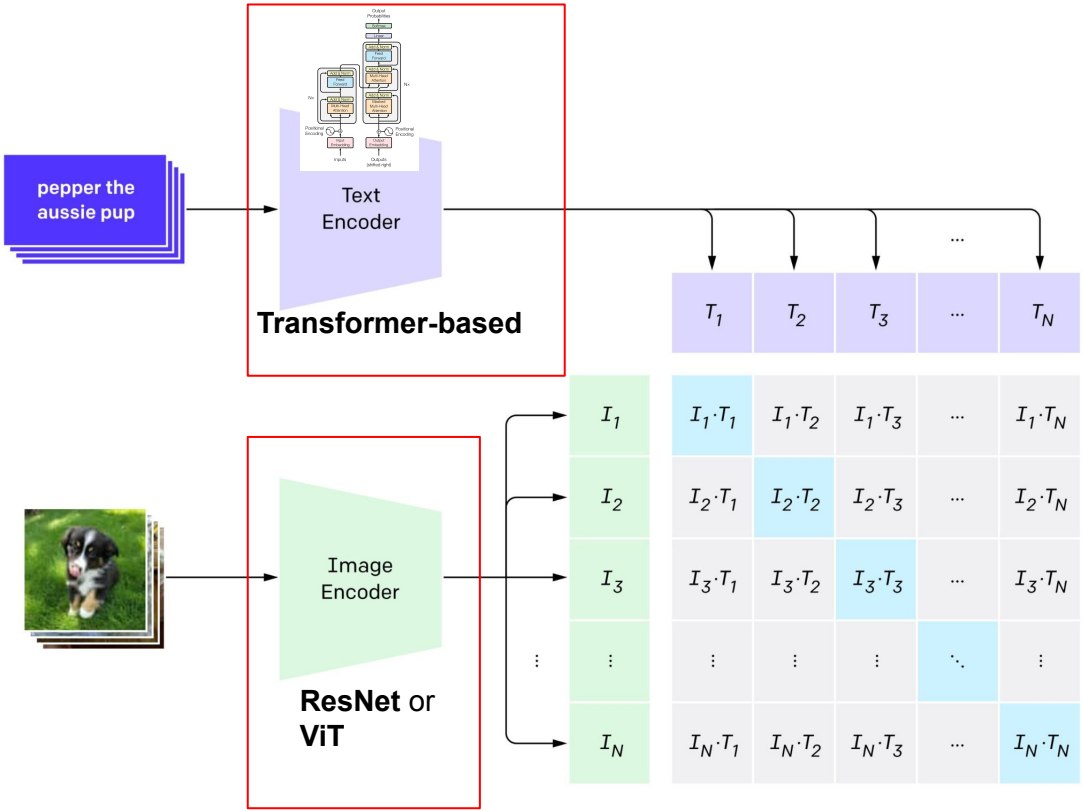
Sample Use Cases:

- **Image Captioning**
- **Visual Question Answering (VQA):**
  - Answering questions about images
- **Image-Text Retrieval:**
  - Finding images based on text queries
- **Text-to-Image Generation:**
  - Using prompts to create images

CLIP revolutionized vision-language AI

But how does it align images with text so effectively?

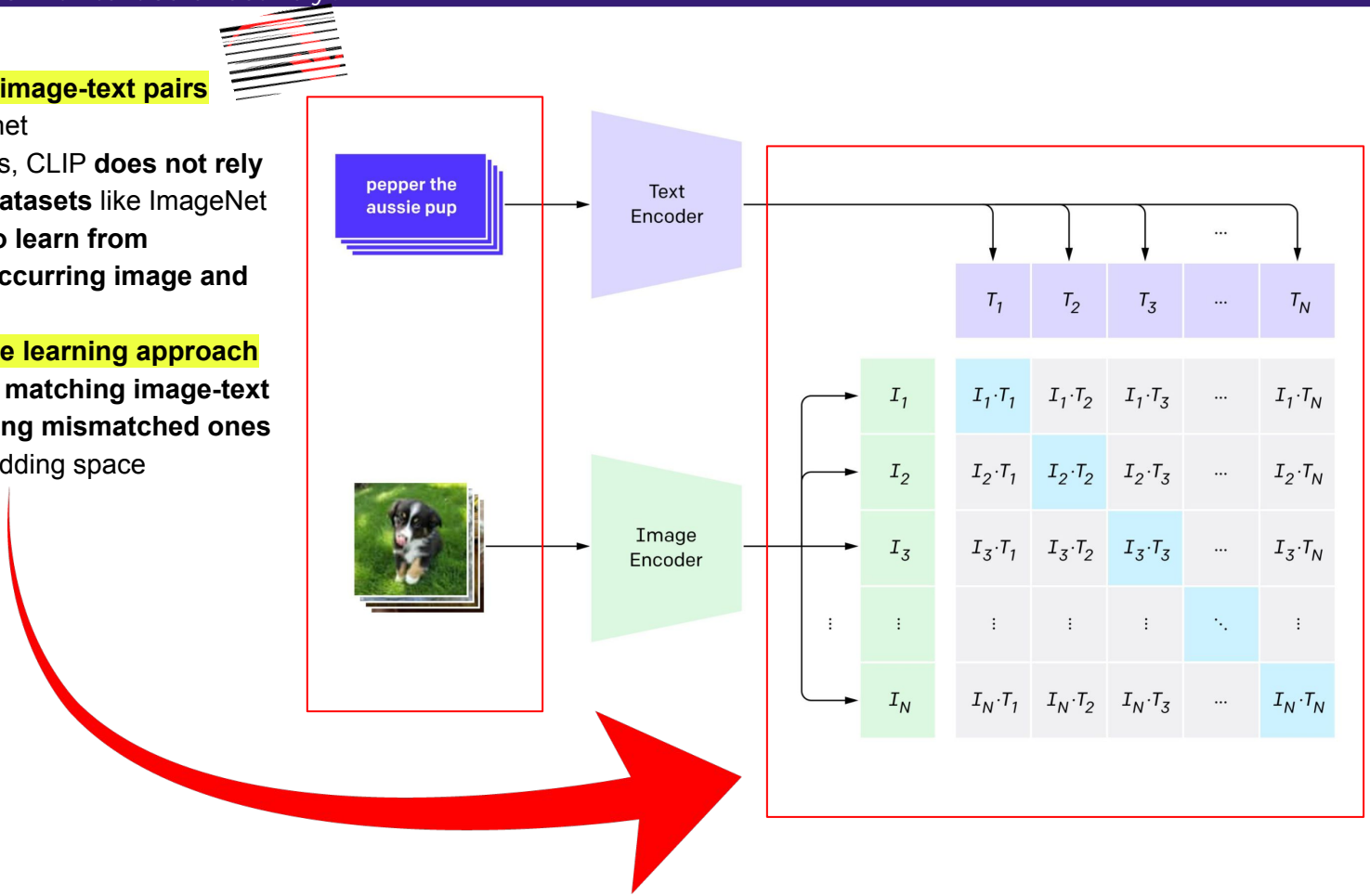
- The two modalities in CLIP are:
  - **Language** - which is transformer-based
  - **Vision** - which is ResNet or ViT



CLIP revolutionized vision-language AI

But how does it align images with text so effectively?

- Trained on **400 million image-text pairs** collected from the internet
- Unlike traditional models, CLIP **does not rely on manually labeled datasets** like ImageNet — it's the **first model to learn from web-scale, naturally occurring image and text data**
- CLIP uses a **contrastive learning approach** — it learns by **bringing matching image-text pairs closer** and **pushing mismatched ones apart** in a shared embedding space



CLIP revolutionized vision-language AI

But how does it align images with text so effectively?

- Trained on **400 million image-text pairs** collected from the internet
- Unlike traditional models, CLIP **does not rely on manually labeled datasets** like ImageNet — it's the **first model to learn from web-scale, naturally occurring image and text data**
- CLIP uses a **contrastive learning approach** — it learns by **bringing matching image-text pairs closer** and **pushing mismatched ones apart** in a shared embedding space

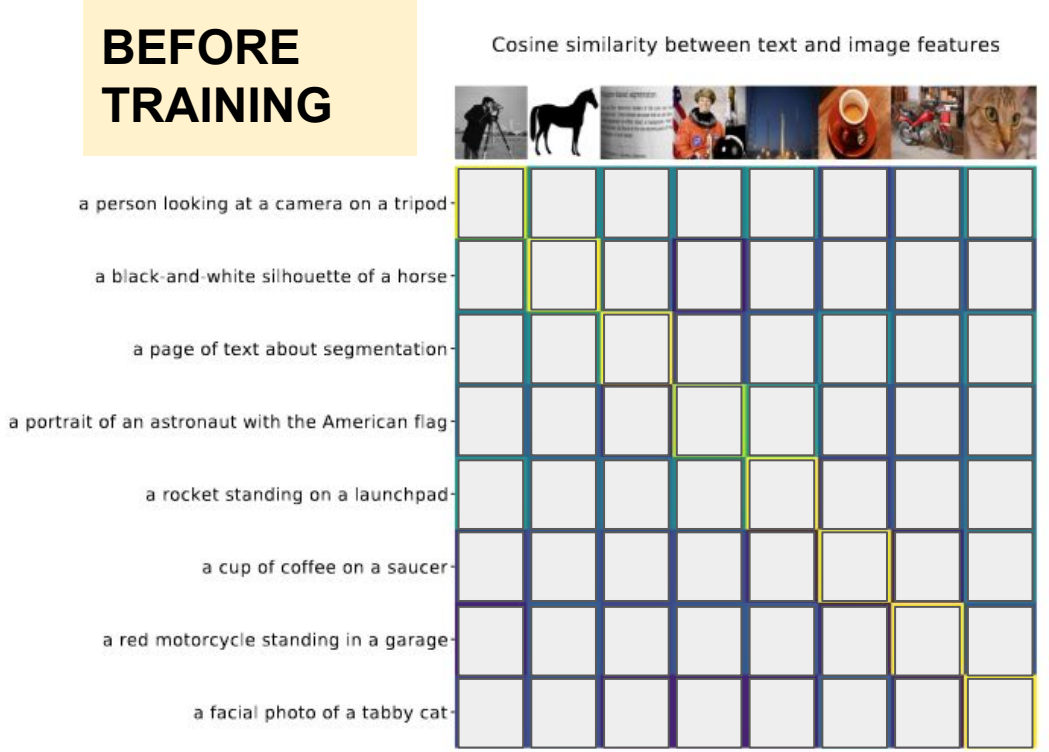


Figure 3: The Cosine similarity matrix between the text and image features pertaining to the `skimage` examples in the *Interacting with CLIP* colab notebook shared by CLIP's authors

CLIP revolutionized vision-language AI

But how does it align images with text so effectively?

- Trained on **400 million image-text pairs** collected from the internet
- Unlike traditional models, CLIP **does not rely on manually labeled datasets** like ImageNet — it's the **first model to learn from web-scale, naturally occurring image and text data**
- CLIP uses a **contrastive learning approach** — it learns by **bringing matching image-text pairs closer** and **pushing mismatched ones apart** in a shared embedding space

AFTER TRAINING

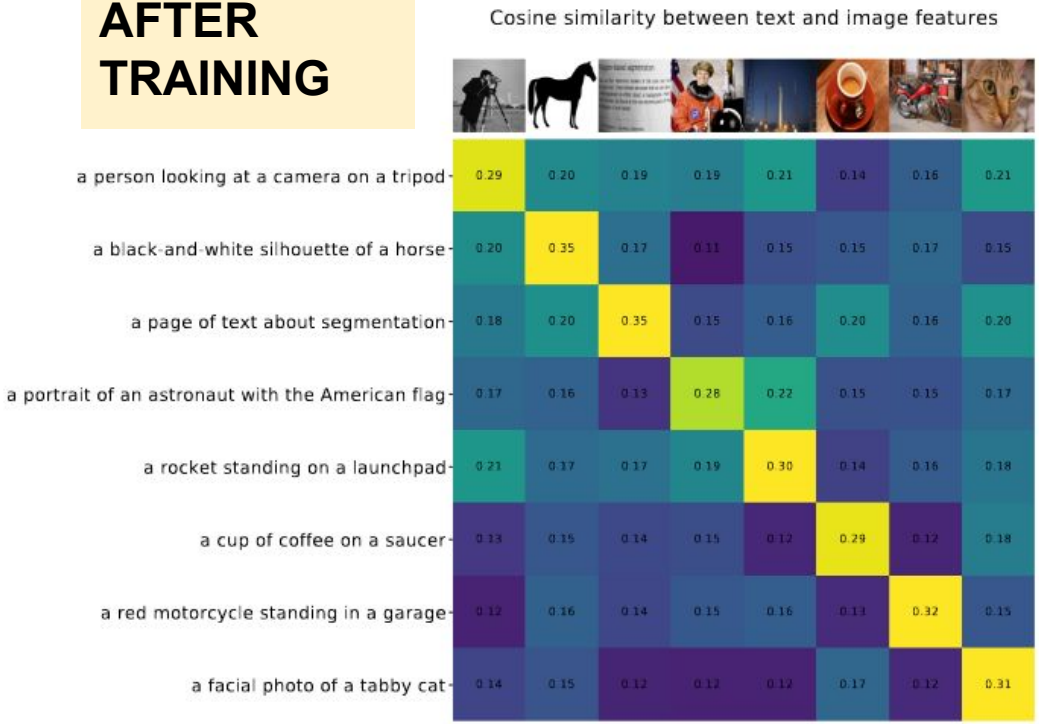


Figure 3: The Cosine similarity matrix between the text and image features pertaining to the `skimage` examples in the *Interacting with CLIP* colab notebook shared by CLIP's authors

VLM Models

	LLM	Vision	VLM
OPEN-SOURCE	**LLaMA 2** (by Meta)		
	OpenLLAMA (by Berkeley AI)		
	Gemma (by Google)		
	Mixtral (by Mistral AI)		
	Mistral (by Mistral AI)	**SAM** (by Meta AI)	**MiniGPT-4**
	Dolly (by Databricks)	**Detectron2** (by Facebook AI R)	**BLIP-2** (by Anthropic)
	Falcon (by TII)	**MMDetection** (by OpenMMlab)	**BLIP** (by Salesforce)
	DeepSeek (by DeepSeek AI)	**YOLOv8** (by Ultralytics)	**CLIP** (by Open AI)
CLOSED-SOURCE	**GPT-4** (by OpenAI)	**Amazon Rekognition** (by AWS)	**GPT-4V** (by OpenAI)
	Claude 3 (by Anthropic)		**Claude 4 Opus** (by Anthropic)
	Gemini (by Google DeepMind)		**Gemini** (by Google DeepMind)
	Command R+ (by Cohere)		**Grok Vision** (by xAI)
	Ernie Bot (by Baidu)		**Imagen**
	Qwen (by Alibaba)		**Midjourney**
	SenseChat (by SenseTime)		**Flux**
	Grok (by xAI)		

VLM Tasks

Discriminative

Understanding Tasks

1. Image-Text Retrieval
🧠 Example: “Find an image that matches the phrase ‘a dog wearing sunglasses.’”

2. Visual Question Answering
🧠 Example: “How many people are in the image?” → “Three”

3. Image Captioning
🧠 Example: Input: 🖼️ → Output: “A cat sitting on a laptop.”

4. Visual Grounding / Referring Expression Comprehension
🧠 Example: “Where is the red backpack.”

5. Multimodal Classification
🧠 Example: “Does this X-ray indicate pneumonia?”

Cognitive

Reasoning Tasks

6. Image-Text Matching
Determine whether a caption correctly describes an image.
🧠 Binary classification: "Yes" or "No"

7. Visual Commonsense Reasoning (VCR)
Given an image, question, and answer choices, select the most plausible one and justify it.
🧠 Example: “Why is the man running?” → “To catch the bus.”

8. Multimodal Chain-of-Thought Reasoning
Provide step-by-step logical reasoning across modalities.

Generative

Generative Tasks

9. Text-to-Image Generation
Generate an image from a textual prompt.
🧠 Example: “A futuristic city on Mars during sunset.”

10. Image Inpainting / Editing
Modify an image based on a text prompt.
🧠 Example: “Remove the tree and add a bench.”

11. Image Captioning with Style or Persona
Describe images with a tone/personality (e.g., sarcastic, poetic, child-like).

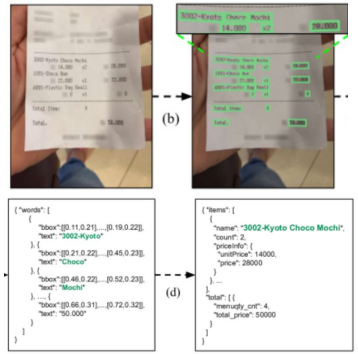
Utility

Utility Tasks

12. Document Visual QA / OCR-based Reasoning
Analyze documents (like receipts, invoices, or tables) using both layout and text.
🧠 Example: “What is the total price on the receipt?”

13. Chart QA / Figure Interpretation
Answer questions about bar charts, pie charts, line graphs, etc.

14. Multimodal Translation
Translate text within images from one language to another.



Yolo-World

https://huggingface.co/spaces/stevengrove/YOLO-World

Improving Open-Vocabulary Object Detection in a Vision Language Model

Anonymous ACCV 2024 Submission

Paper ID #94

Abstract. Vision Language Models (VLMs) are foundational models that provide meaningful answers to questions requiring image understanding. One of the many important tasks that VLMs can perform is open-vocabulary object detection (OVD), which involves detecting objects beyond the known classes the model was trained on. Despite training on extensive vision and language datasets, state-of-the-art (SOTA) VLMs such as LLaVA-v1.5, LLaVA-NEXT, and Qwen-VL-Chat achieve only 4.8, 14.4, and 18.6 AP respectively on the LVIS novel object detection benchmark. In this paper, we propose improving the OVD capability of LLaVA-v1.5 by fine tuning it on a new dataset generated automatically by FireLLaVA VLM and YOLO-World OVD. Our model, called LLaVA-World, achieves 25.0 AP on LVIS, a score that is state-of-the-art among VLMs and comparable to SOTA specialist OVD models such as OWL-ViT (25.6 AP) and YOLO-World (27.6 AP). LLaVA-World achieves this strong OVD score while maintaining the overall LLaVA performance on general language-vision tasks.

Keywords: OVD · Open-Vocabulary Object Detection · Vision Language Model

Human: What is going on here?

LLaVA-v1.5-13B: In the image, a man is driving a horse-drawn carriage down a city street. The carriage is pulled by a white horse, and the man is sitting on the carriage, guiding it. The scene appears to be set in an old city, possibly in a foreign country, with a building visible in the background. The man is also wearing a cowboy hat, which adds to the unique atmosphere of the scene. The horse-drawn carriage is a traditional mode of transportation that is not commonly seen in modern cities, making this image quite captivating and nostalgic.

Fig. 1: VLMs are capable of answering questions that require understanding the context of the image. In this case, a vague or ambiguous question can not be answered by LLM itself without "seeing" the image.

Fig. 2: ChatGPT-4o offers a poor object detection estimate on rare object `horse buggy`.

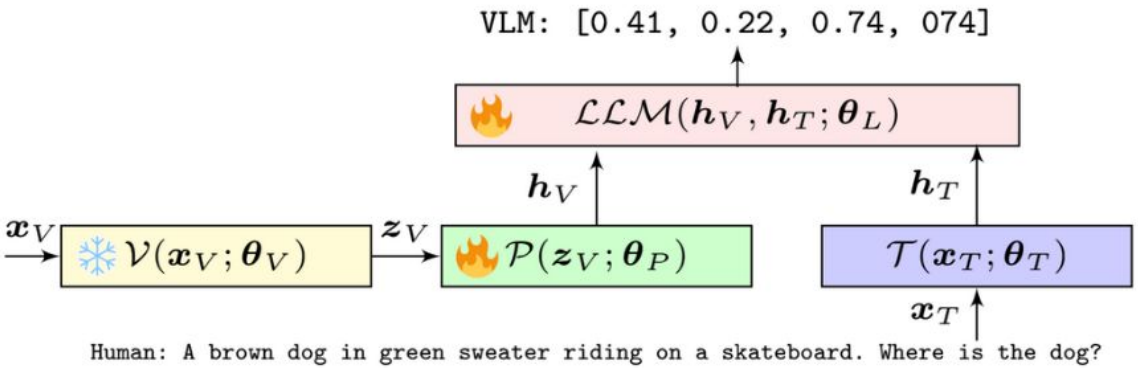
LLaVa-World

(a) LLaVA-v1.5-13B (b) LLaVA-World-13B (Ours)

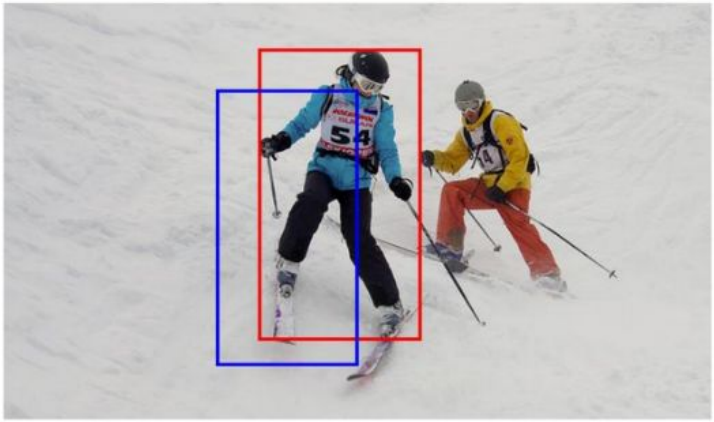
Fig. 3: Comparison of novel object detection results between LLaVA-v1.5-13B and LLaVA-World-13B for rare object `horse buggy`.

LLaVa-World

Table 1: LLaVA-World fine-tuning recipe with a sample multi-turn conversation between human and VLM.



Role	Conversation
Human	A brown dog in green sweater riding on a skateboard. Where is the dog?
VLM	[0.41, 0.22, 0.74, 074]
Human	A cute little dog is on a skateboard. Locate the skateboard.
VLM	[0.37, 0.69, 0.89, 0.89]
Human	A small brown dog riding a skateboard with their owner. Find the cat.
VLM	There is no cat in the image.



Human: Locate the woman in a blue jacket skiing in the image. Output must be bounding box coordinates with the format [xmin, ymin, xmax, ymax].

LLaVA-v1.5^{7B}: [0.30, 0.21, 0.50, 0.87]

LLaVA-World^{7B}: [0.34, 0.13, 0.60, 0.85]

Fig. 6: An example of a referring expression comprehension evaluation using LLaVA-v1.5^{7B} (**blue**, vanilla) and LLaVA-World^{7B} (**red**, ours).

LLaVa-World

Fig. 7: Examples of annotated images with bounding boxes queried from LLaVA-v1.5^{13B} (**blue**, vanilla) and LLaVA-World^{13B} (**red**, ours).

Nikko Carlo Yabut

Hearing Machines

An Introduction to Audio AI

From Sound to Signal – The Basics of Audio AI

- Key Concepts:
  - Audio = Time-series data (waveforms)
  - Spectrograms = Images of sound
  - Sampling rate (how often we record sound)
  - Features: MFCCs, Mel-Spectrograms

Sample Use Cases:

- **Audio Captioning**
  - describe what's heard
- **Spoken QA**
  - ask questions about spoken content
- **Voice Style Transfer**
  - Making one person sound like another
- **Multimodal assistants**
  - voice + vision + text

Audio AI Models

OPEN-SOURCE

LLM

- **LLaMA 2** (by Meta)
- **OpenLLAMA** (by Berkeley AI)
- **Gemma** (by Google)
- **Mixtral** (by Mistral AI)
- **Mistral** (by Mistral AI)
- **Dolly** (by Databricks)
- **Falcon** (by TII)
- **DeepSeek** (by DeepSeek AI)

Vision

- **SAM** (by Meta AI)
- **Detectron2** (by Facebook AI R)
- **MMDetection** (by OpenMMlab)
- **YOLOv8** (by Ultralytics)

VLM

- **MiniGPT-4**
- **BLIP-2** (by Anthropic)
- **BLIP** (by Salesforce)
- **CLIP** (by Open AI)

Audio

- **ESPnet**
- **Wav2Vec**
- **Whisper**

CLOSED-SOURCE

LLM

- **GPT-4** (by OpenAI)
- **Claude 3** (by Anthropic)
- **Gemini** (by Google DeepMind)
- **Command R+** (by Cohere)
- **Ernie Bot** (by Baidu)
- **Qwen** (by Alibaba)
- **SenseChat** (by SenseTime)
- **Grok** (by xAI)

Vision

- **Amazon Rekognition** (by AWS)

VLM

- **GPT-4V** (by OpenAI)
- **Claude 4 Opus** (by Anthropic)
- **Gemini** (by Google DeepMind)
- **Grok Vision** (by xAI)
- **Imagen**
- **Midjourney**
- **Flux**

Audio

- **ElevenLabs**
- **Google Speech-to-Text**
- **Amazon Transcribe**
- **Microsoft Azure Speech**
- **Descript Overdub**

Audio AI Tasks

Speech Tasks

Automatic Speech Recognition (ASR)
Converting spoken language into written text
Example: Transcribing podcasts or voice memos
Models: Whisper, Wav2Vec 2.0

Text-to-Speech (TTS)
Generating speech from text input
Example: Screen readers, virtual assistants
Models: Tortoise TTS, ElevenLabs

Speaker Diarization
Determining "who spoke when" in an audio stream
Example: Meeting transcription with speaker labels

Speaker Identification / Verification
Recognizing or confirming speaker identity
Example: Voice authentication for security

Voice Cloning
Creating synthetic voices based on a few samples of a real voice
Example: Personal voice assistants, voice preservation

Sound Processing

Audio Classification
Classifying audio into categories (e.g., music genre, environmental sound)
Example: Detecting dog barks vs. sirens in urban monitoring

Sound Event Detection (SED)
Locating and identifying specific sound events in an audio stream
Example: Identifying gunshots, baby cries, or alarms in surveillance

Acoustic Scene Classification
Recognizing the type of environment from sound (e.g., street, forest, cafe)
Example: Smart home or robotics context awareness

Emotion Recognition from Speech
Detecting emotional tone (e.g., happy, angry, sad)
Example: Call center analytics, mental health monitoring

Audio Denoising / Enhancement
Removing background noise or enhancing clarity
Example: Improving call quality, podcast editing

Generative

Music Generation / Composition
Creating original music or accompanying melodies
Example: AI-generated background music
Models: MusicLM, Riffusion

Voice Style Transfer / Singing Voice Conversion
Changing the style or identity of a voice without altering content
Example: Making one person sound like another

Audio-to-Audio Translation
Converting one type of audio to another (e.g., humming to instruments)

Multimodal Audio Tasks
Combining audio with other modalities (e.g., lip reading, audio-visual speech recognition)
Example: Video subtitles from lip + sound input

ESPnet
Wav2Vec
Whisper

Audio

ElevenLabs
Google Speech-to-Text
Amazon Transcribe
Microsoft Azure Speech
Descript Overdub

Models

	LLM	Vision	VLM	Audio
OPEN-SOURCE	LLaMA 2 (by Meta)			
	OpenLLAMA (by Berkeley AI)			
	Gemma (by Google)			
	Mixtral (by Mistral AI)			
	Mistral (by Mistral AI)	SAM (by Meta AI)	MiniGPT-4	
	Dolly (by Databricks)	Detectron2 (by Facebook AI R)	BLIP-2 (by Anthropic)	ESPnet
	Falcon (by TII)	MMDetection (by OpenMMlab)	BLIP (by Salesforce)	Wav2Vec
	DeepSeek (by DeepSeek AI)	YOLOv8 (by Ultralytics)	CLIP (by Open AI)	Whisper
CLOSED-SOURCE	GPT-4 (by OpenAI)	Amazon Rekognition (by AWS)	GPT-4V (by OpenAI)	ElevenLabs
	Claude 3 (by Anthropic)		Claude 4 Opus (by Anthropic)	Google Speech-to-Text
	Gemini (by Google DeepMind)		Gemini (by Google DeepMind)	Amazon Transcribe
	Command R+ (by Cohere)		Grok Vision (by xAI)	Microsoft Azure Speech
	Ernie Bot (by Baidu)		Imagen	Descript Overdub
	Qwen (by Alibaba)		Midjourney	
	SenseChat (by SenseTime)		Flux	
	Grok (by xAI)			

Multimodal Mini-hackathon

Instruction:
In your assigned group, brainstorm and propose a business use case that creatively integrates multimodal AI—incorporating language, vision, and audio capabilities — to solve a real-world marketing problem.

Group 1	Group 2
James	Noela
Cedric	Justin
Junielle	Cris
Bryan	Carl
Rexie	Pauline
Clarence	Xy

Expected output

- Language, Audio, Vision, Video, etc…
- Landing page - use bolt.new
- Streamlit app - for basic functionality
- Pitch deck -
  - ChatGPT to create narrative
  - Gamma to create deck

- Presentation: 3:30PM

Resources:

- Multimodal streamlit playground deployment guide

  Github:
  Week 4 - Multimodal Capabilities/🚀 How to Deploy in Streamlit.pdf

Nikko Carlo Yabut

Foundations of Embeddings

Task Adaptation

OPEN-SOURCE

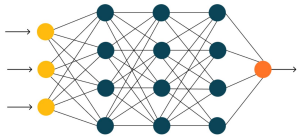
- **LLaMA 2** (by Meta)
- **OpenLLAMA** (by Berkeley AI)
- **Gemma** (by Google)
- **Mixtral** (by Mistral AI)
- **Mistral** (by Mistral AI)
- **Dolly** (by Databricks)
- **Falcon** (by TII)
- **DeepSeek** (by DeepSeek AI)

LLM

CLOSED-SOURCE

- GPT-4 (by OpenAI)
- Claude 3 (by Anthropic)
- Gemini (by Google DeepMind)
- Command R+ (by Cohere)
- Ernie Bot / Ernie 4.0 (by Baidu)
- Qwen (by Alibaba)
- SenseChat (by SenseTime)
- Grok (by xAI)

Model



Allows alteration of model's **weights**, **behavior**, or **training** data.

- Full Training
- Partial Training (QLoRA)
- RLHF (Reinforcement Learning with Human Feedback)

Prompt

User: "what hafen vella?"

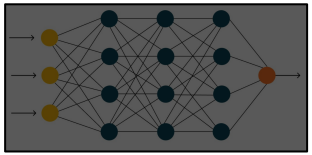
Output

ChatGPT: The phrase "What hafen Vella?" is a viral meme that originated from a 2013 episode of It's Showtime…

Prompting Techniques
- Zero-shot Prompting
- One-shot Prompting
- Few-shot Prompting
- …

Prompt Frameworks
- RICCE
- PACE
- APE

Model



Weights, **behavior**, or **training** CANNOT be altered.

a.k.a. "Black Boxes"

Prompt

Output

Prompting Techniques / Frameworks
*same as above

What's on your mind today?

If you were to build a chatbot that could answer questions about your company's documents, how would you do it?

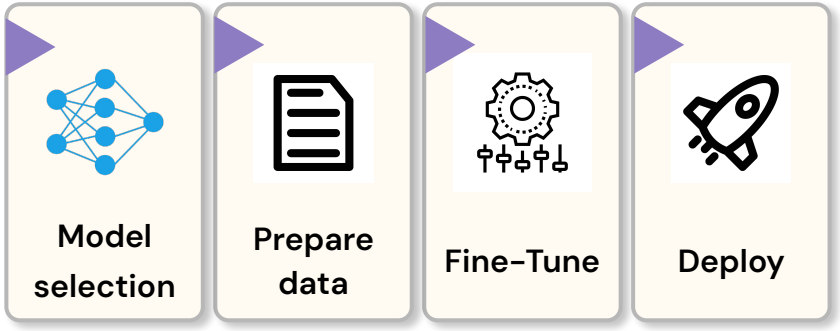
+ Tools

Fine-tuning as default strategy

"Take a base LLM like LLaMA, Mistral, or GPT-3. We'll just fine-tune it with our company's documents to make it smart."

Problems With Fine-tuning

General steps in Fine-tuning



COST
- Fine-tuning large models is expensive. Even open-source require GPU hours, storage, etc.

RIGIDITY
- Once a model is fine-tuned on a snapshot of your data, it doesn't generalize well to new content.

REPETITION
- Every time your knowledge base changes (e.g., new products, policies), you need to fine-tune again.

VERSIONING
- You end up managing multiple model versions per department, per document version, etc.

SLOW ITERATION
- Even simple updates (e.g., adding a new FAQ) require model retraining. This breaks rapid prototyping.

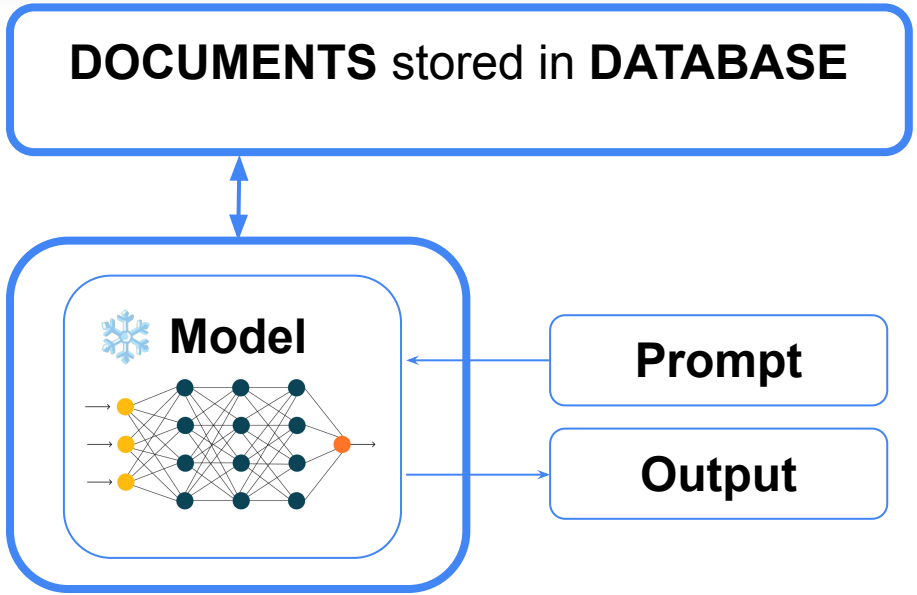
What's on your mind today?

If you were to build a chatbot that could answer questions about your company's documents, how would you do it?

+ Tools

Better Approach:

"Instead of fine-tuning the model every time the knowledge base updates, what if we just store the documents in a database — and let the model retrieve the relevant chunks during inference?"



What's on your mind today?

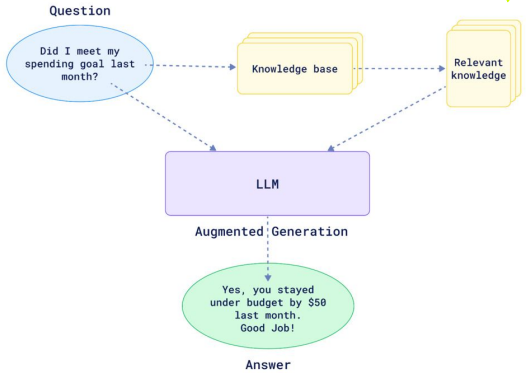
If you were to build a chatbot that could answer questions about your company's documents, how would you do it?

+ Tools

Better Approach:

"Instead of fine-tuning the model every time the knowledge base updates, what if we just store the documents in a database — and let the model retrieve the relevant chunks during inference?"

RAG



- Use a frozen, pretrained model (e.g., GPT or Mistral)
- Store your documents in a **vector database** (like FAISS, Weaviate, Qdrant, etc.)
- At runtime, convert the query and your documents into **embeddings** (aka latent representations)
- Retrieve relevant documents using **semantic similarity**
- Feed them into the LLM (via context window)

What's on your mind today?

If you were to build a chatbot that could answer questions about your company's documents, how would you do it?

+ Tools

Better Approach:

"Instead of fine-tuning the model every time the knowledge base updates, what if we just store the documents in a database — and let the model retrieve the relevant chunks during inference?"

RAG

"So instead of adapting the model to the documents, we adapt the documents to the model — using embeddings.

RAG

EMBEDDINGS?

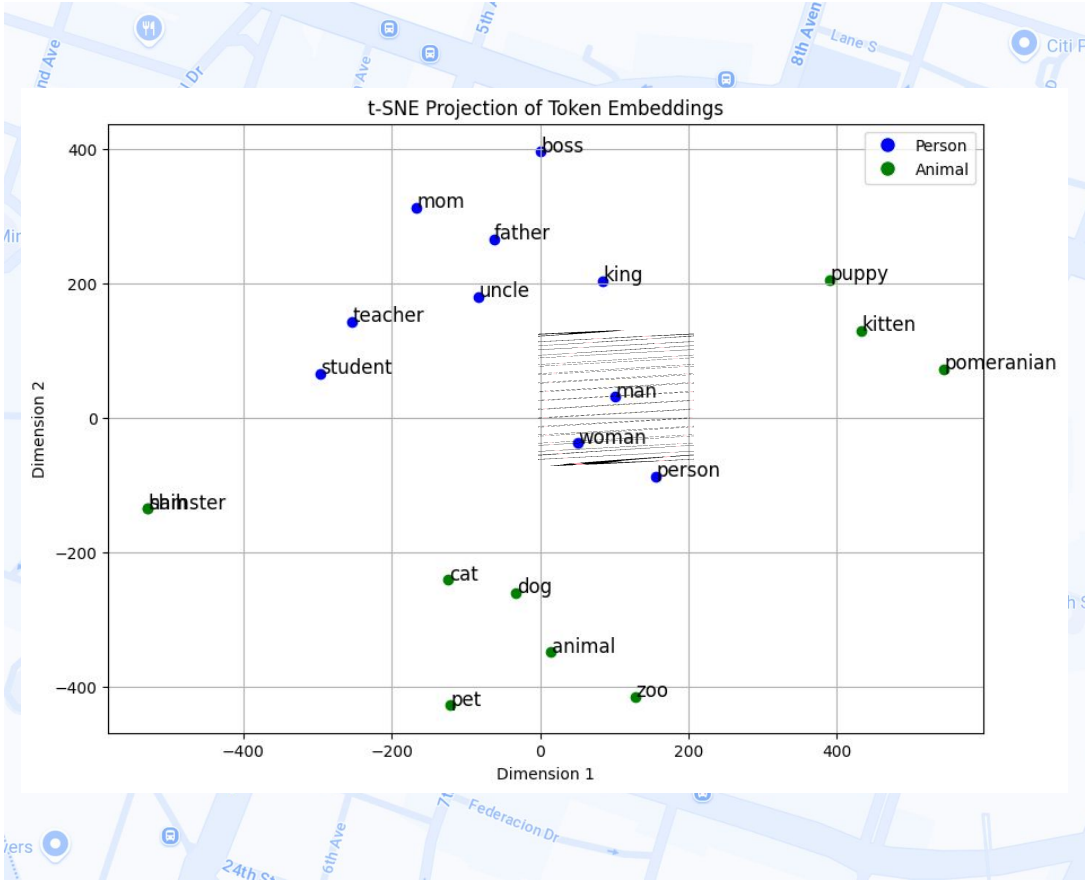
“So instead of adapting the model to the documents, we adapt the documents to the model — using embeddings.

Embeddings represent words through bunch of numbers

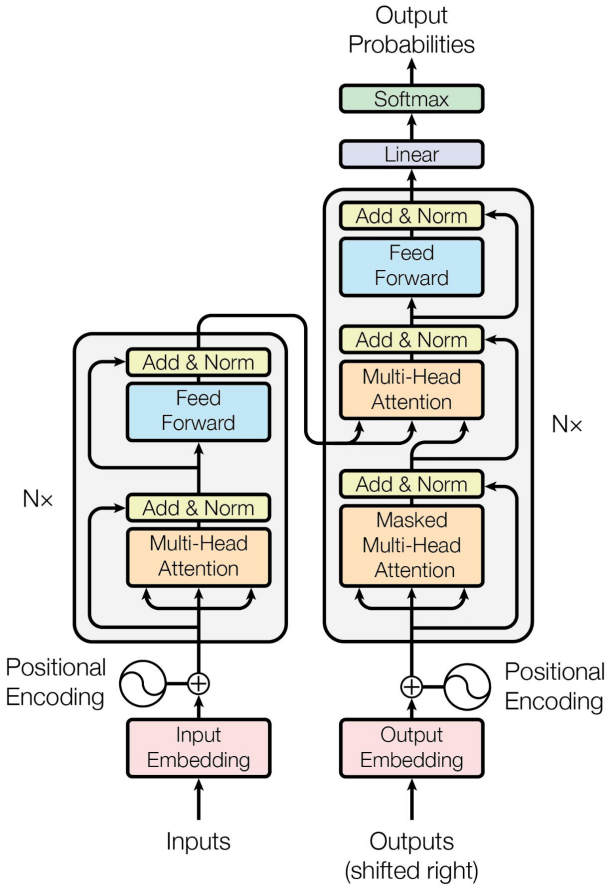
Think of it like this:

If words were cities on a map, embeddings are the coordinates.

Words with similar meanings, like ‘man’ and ‘woman’, end up close to each other on that map.

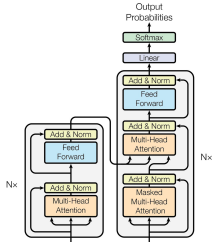


How are **Embeddings** created? (via transformers)



https://www.datacamp.com/tutorial/how-transformers-work

How are **Embeddings** created? (via transformers)



How are you?

INPUT

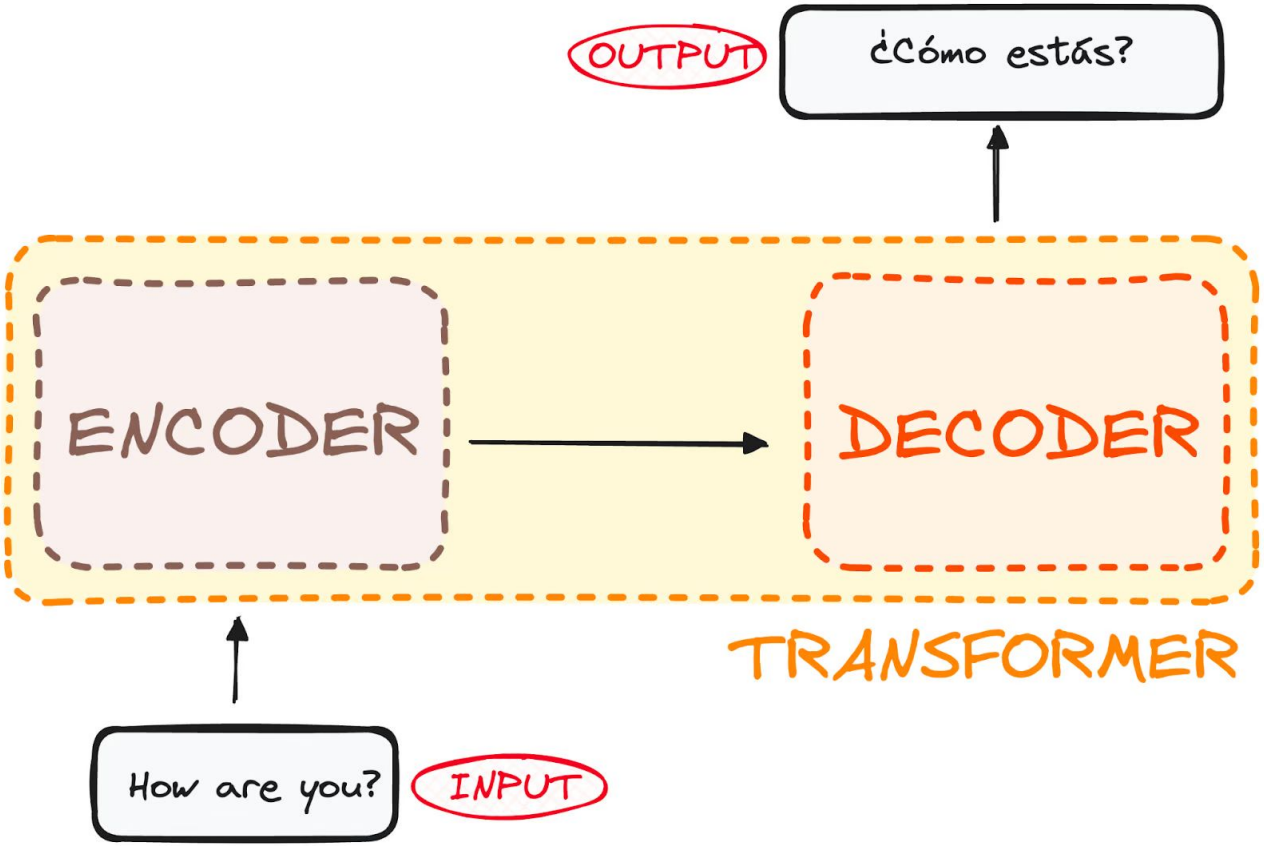
TRANSFORMER

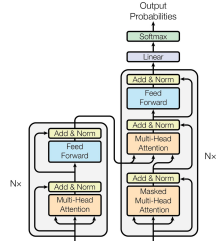
¿Cómo estás?

OUTPUT

https://www.datacamp.com/tutorial/how-transformers-work

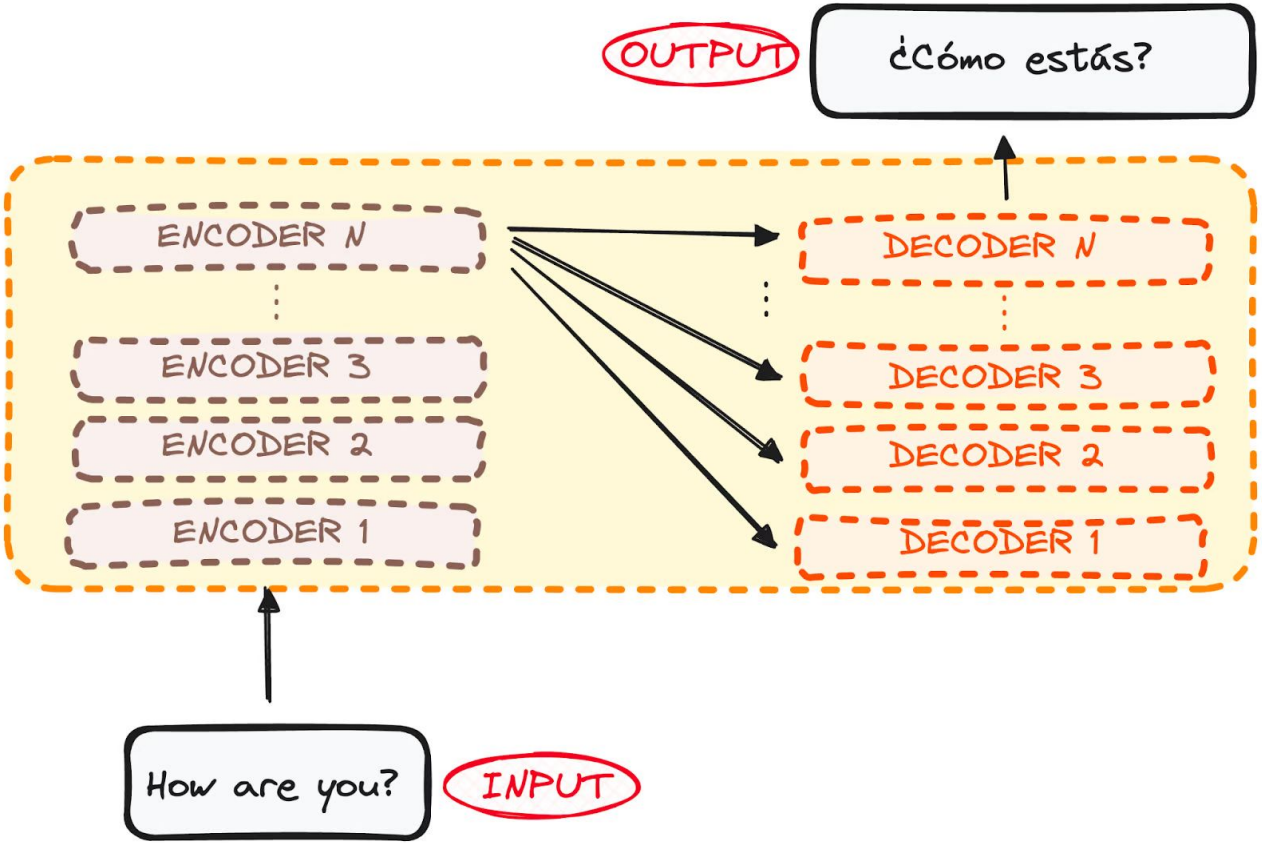
How are **Embeddings** created? (via transformers)

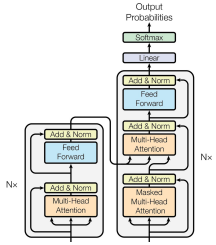




https://www.datacamp.com/tutorial/how-transformers-work

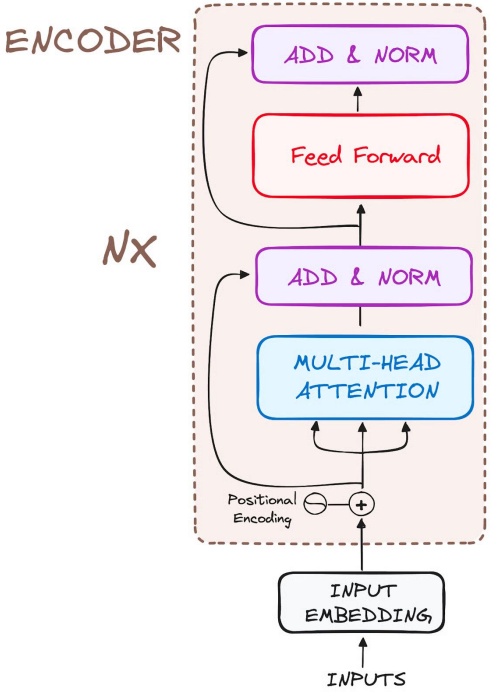
How are **Embeddings** created? (via transformers)

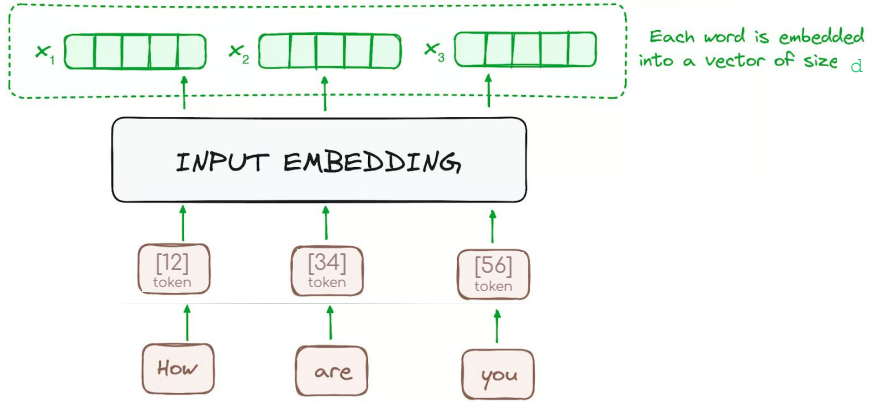


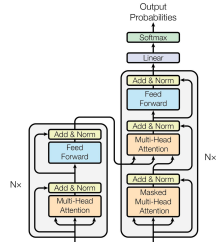


https://www.datacamp.com/tutorial/how-transformers-work

How are **Embeddings** created? (via transformers)







https://www.datacamp.com/tutorial/how-transformers-work

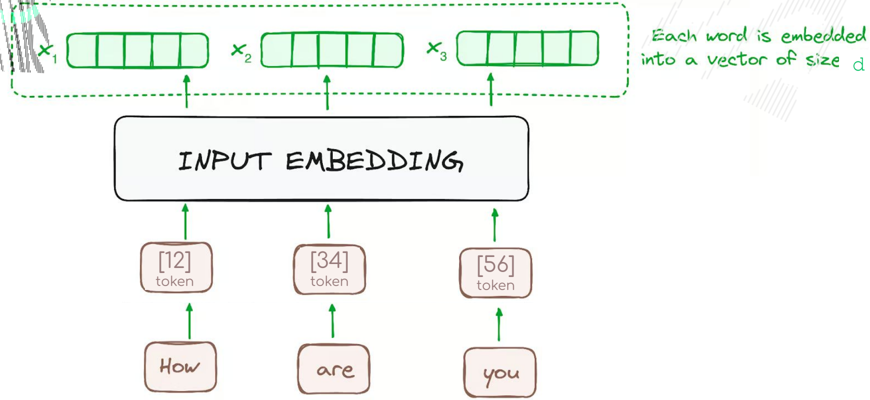
How are **Embeddings** created? (via transformers)

📌 Word: How
🧩 Tokenized: ['how']
🔢 Token ID: 2129

🔷 Static Embedding Vector for 'How' (shape: (768,)):


```
[-2.40632426e-02  3.17239240e-02 -7.56581724e-02  6.26911037e-03 -2.58011918e-04 -4.89562452e-02 -8.62864181e-02 -1.48444436e-02  2.95870882e-02 -2.93501709e-02 -1.86109450e-02 -2.58933883e-02  4.38467599e-02
  5.06787337e-02 -2.69273855e-02 -9.60835367e-02  2.47513223e-02 -1.66852530e-02  4.85536605e-02  1.61708891e-02 -1.12872068e-02  2.37299539e-02 -8.33070744e-03 -3.69310901e-02 -2.43126173e-02 -5.99710383e-02
 -1.38285588e-02 -7.60446936e-02  7.07411915e-02 -2.73569897e-02 -3.67836654e-02 -7.62912259e-02 -8.92018452e-02  6.45733774e-02 -3.21142282e-03 -3.13841812e-02 -7.32790530e-02 -2.19877567e-02 -4.54654545e-03
 -5.16635329e-02 -8.74689315e-03  1.45732490e-02 -8.29743370e-02  6.72775134e-02  3.09977941e-02  4.56132321e-03  2.29086820e-02 -3.61224562e-02  3.17416489e-02  1.83614213e-02 -4.02215775e-03 -4.04329039e-02
  3.96960974e-03  5.34170587e-03 -1.11106066e-02  9.45139024e-03 -4.14175689e-02  1.76874711e-03 -1.97757743e-02 -2.28614025e-02 -9.49357823e-03  8.20351951e-03 -1.82070285e-02 -3.47239673e-02  4.16595377e-02
 -5.05649485e-02 -8.35249200e-03  1.25095462e-02  4.21202416e-03 -2.61297952e-02 -7.05806613e-02 -2.02179626e-02 -3.15761529e-02 -4.31864224e-02  3.85294226e-03  7.09294435e-03 -3.02653443e-02  1.60581004e-02
  4.92024329e-03 -2.77611353e-02  3.49431969e-02  2.80356519e-02  3.50493006e-02  2.73176637e-02 -1.19978096e-02 -8.49911273e-02 -4.09458280e-02 -2.72356793e-02 -7.49194324e-02 -7.59286061e-02  4.99165803e-02
 -4.78370413e-02 -6.51342273e-02 -3.75630707e-02 -3.04567311e-02 -6.00833967e-02 -5.35118720e-03 -1.03650177e-02 -3.19029652e-02 -3.16432454e-02  4.52513341e-03  6.43489733e-02 -8.88413377e-03 -9.86471213e-03
 -1.52054569e-02 -2.22128467e-03  8.31383560e-03 -2.57547889e-02 -1.01689184e-02 -8.10694299e-04 -6.42709807e-02 -4.49671820e-02 -3.81466933e-02 -4.20909375e-02 -4.54924069e-02  7.92594850e-02  8.85786768e-03
  1.45143382e-02 -1.64769571e-02 -1.84432361e-02 -7.20525673e-03  2.29655154e-04  1.04185855e-02  1.79463569e-02  8.97494785e-04 -5.32222316e-02 -1.26120383e-02 -5.76113574e-02 -1.71773471e-02  2.03900337e-02
 -1.71129815e-02  8.24113283e-03  2.27314793e-03  6.85477406e-02 -6.80906326e-02  3.66026238e-02  3.42215300e-02  2.34122779e-02 -4.25386056e-02 -6.82519702e-03  3.85989621e-02  7.46158464e-03  3.85151245e-02
 -3.69149111e-02  3.06216143e-02 -7.47312456e-02 -1.02839796e-02 -3.08722034e-02  7.31505966e-03 -8.44044462e-02  4.60562184e-02  2.53819246e-02 -5.24426326e-02 -2.34480351e-02 -2.61210948e-02 -2.73317080e-02
  2.94399988e-02 -2.03207750e-02 -2.91144606e-02  2.75079440e-02 -5.10416701e-02  5.68141742e-03 -3.95116992e-02  9.67309996e-03  4.10742068e-05 -7.00547323e-02 -5.13092540e-02 -5.24353981e-02  1.59527287e-02
  4.54671569e-02  9.29208938e-04 -6.32209256e-02 -3.60561609e-02  1.03921546e-02 -3.00257392e-02 -1.52557818e-02  8.80255736e-03  5.90848504e-03 -3.73701751e-02 -1.89466793e-02 -4.74576699e-03 -5.00825047e-02
 -1.03363851e-02 -3.23428623e-02  1.14934720e-01  2.80265342e-02 -6.46438356e-03 -7.34714642e-02  8.06048587e-02 -4.89006229e-02  2.72423513e-02  2.29129428e-03 -5.01409620e-02 -5.81299774e-02 -1.98586285e-02
 -2.16328800e-02 -2.25455519e-02 -1.76306274e-02  2.31720675e-02  3.54568698e-02 -1.79572664e-02  2.39172839e-02 -1.86421890e-02 -7.19889179e-02 -1.86346546e-02  3.24567296e-02 -5.17822616e-02  1.19499033e-02
  2.79369392e-02 -6.09460287e-02 -4.28852178e-02  2.25416645e-02 -2.86062695e-02 -3.66943628e-02  3.00906040e-02 -3.10584940e-02 -6.72834590e-02 -4.79315780e-02  3.69124045e-03 -2.50349734e-02  4.02781442e-02
  7.43498141e-03 -5.60195372e-02 -1.35053573e-02 -3.45139615e-02  3.79546955e-02  5.51038310e-02 -3.35163772e-02  5.03432900e-02 -9.50830814e-04 -2.85925046e-02  1.77440383e-02  1.03203962e-02 -1.42628606e-02
  1.98729225e-02  4.85438630e-02  4.03669588e-02  6.81296410e-03 -8.47603157e-02  1.96953397e-02  6.08333526e-03  2.46805009e-02 -3.51215117e-02 -3.49530317e-02 -1.43865198e-02  3.60011533e-02 -2.97866575e-02
 -2.72921775e-03 -1.59283988e-02 -9.03266203e-03 -4.00798805e-02 -3.95079665e-02  1.17932884e-02 -2.03591753e-02  1.05042495e-02 -2.03903019e-02 -1.80941001e-02 -3.59604321e-02 -9.18738842e-02  6.09645434e-02
  1.80536543e-03  2.56685866e-03 -3.29396017e-02  1.34370727e-02  1.21362126e-02 -3.67373712e-02 -8.94654635e-03 -4.06659022e-02  2.78272592e-02  9.23155714e-03  1.16159860e-02 -4.49797474e-02  1.41576407e-02
 -1.11131638e-01  1.03015658e-02 -4.55153082e-03 -1.28376996e-02 -7.35259429e-02  2.52520982e-02 -2.76803765e-02 -3.90702896e-02 -3.61095928e-02 -4.60212603e-02  1.90247912e-02  4.47232239e-02 -1.42136076e-02
  1.01556890e-02  2.36594882e-02 -4.32280488e-02  1.24429623e-02 -4.39441130e-02 -1.82906911e-02  4.15928699e-02 -6.13628216e-02  4.86430107e-03  2.65281480e-02 -3.06413067e-03 -4.58845543e-03  1.36416610e-02
 -5.12329116e-02  4.49450947e-02 -3.06420941e-02  3.11115067e-02  5.19374460e-02 -3.73131372e-02 -4.36187163e-02  1.31423762e-02 -7.99829606e-03 -6.31169006e-02 -2.69059557e-02 -1.93704180e-02 -2.04635691e-03
  8.94358009e-03 -6.86627105e-02 -6.04309589e-02  2.33461652e-02 -7.06690596e-04 -4.44739945e-02  1.83669981e-02 -1.50816711e-02 -5.70608526e-02 -6.09482639e-04 -2.96346750e-02  1.66279897e-02  4.40274552e-02
 -7.93768093e-03  2.15591900e-02 -8.25689435e-02  1.33756967e-02 -4.13693562e-02 -4.56367712e-03 -7.26461969e-03  2.36548739e-03 -1.13174543e-01 -8.58894959e-02 -3.93058993e-02  5.97350579e-03 -3.45391035e-02
  3.41741815e-02 -9.25091878e-02 -5.73617369e-02 -3.12605239e-02 -8.67043156e-03 -5.94054768e-03  1.92349888e-02  9.78673249e-03  7.47569501e-02 -3.05492338e-02 -1.03461161e-01  5.84605336e-02  1.74750816e-02
 -5.44028766e-02  4.93369475e-02 -1.96900498e-02  9.62559320e-03 -4.25645970e-02 -8.98296013e-02 -5.03054932e-02 -5.04233837e-02 -6.15962669e-02 -3.69961634e-02 -2.78792251e-02  8.13827757e-03  3.73848947e-03
  5.26996851e-02  1.62710864e-02  5.72457677e-04 -3.81550156e-02 -4.81124362e-03 -9.95321944e-02  1.06638689e-02 -3.81066315e-02 -2.29458977e-02 -6.60021529e-02  3.84338684e-02 -3.70474532e-02 -1.22215413e-03
  6.31565526e-02  1.39290588e-02  2.92073451e-02  3.00167818e-02 -7.62918293e-02 -2.23823953e-02 -6.00151010e-02  3.99245955e-02 -5.81338024e-03 -1.29170343e-02  2.33683120e-02 -3.22067700e-02 -2.15006135e-02
 -7.66447410e-02 -3.54666859e-02 -9.19029675e-03 -2.27978323e-02 -2.17447542e-02 -4.49042954e-02  9.99037363e-03 -2.96426043e-02 -2.87224278e-02 -2.63543427e-02 -3.99767496e-02 -2.76514534e-02 -4.02079299e-02
 -6.59588277e-02  6.84235394e-02 -8.39904472e-02 -2.62792427e-02  1.81724560e-02 -1.36873908e-02 -4.11824621e-02 -3.11036073e-02 -7.22710509e-03 -4.45440374e-02 -7.06092715e-02 -1.50430547e-02 -9.25701484e-03
 -3.85007672e-02 -1.26395272e-02  3.74264009e-02  4.08810377e-02  2.48979535e-02 -4.75944355e-02 -2.07229517e-02  1.01020150e-02 -4.31836694e-02 -3.27098407e-02  2.57209986e-02 -8.33619386e-03 -2.78881397e-02
  2.67299212e-04  3.83807234e-02 -2.68067524e-04 -3.54761295e-02 -4.68954369e-02 -2.21075229e-02  1.88480578e-02 -6.35658652e-02 -2.07822993e-02 -3.00569627e-02 -2.63915136e-02 -3.51044759e-02 -2.96256281e-02
 -4.72513363e-02 -8.74045957e-03 -5.50780329e-04 -6.05693720e-02 -6.11014515e-02  1.91002674e-02  2.26630848e-02  3.75893265e-02 -2.15478893e-02 -7.14205857e-03  1.22040827e-02  3.28841782e-03 -3.31071988e-02
  1.28081040e-02  3.57418172e-02 -1.59375686e-02 -1.41410902e-03  4.88664806e-02 -9.71987993e-02 -2.06610584e-03 -1.89623013e-02  9.96184722e-03  4.53704782e-02  1.96120664e-02 -1.68459173e-02  1.30380057e-02
 -1.66016687e-02  3.19261774e-02  8.40354525e-03  2.03369930e-03 -8.23270809e-03 -1.20725464e-02 -5.65328747e-02 -1.67112183e-02 -5.16212694e-02  4.16195579e-02  4.16581705e-03 -2.00486090e-03 -4.85344604e-02
 -1.62212793e-02 -6.42460072e-03  3.35895419e-02  2.81633604e-02  2.43080556e-02  2.69450247e-02  1.66514739e-02 -1.75541081e-02  3.39950845e-02 -5.51930964e-02 -3.53590287e-02 -3.94096710e-02 -6.34305254e-02
 -3.63813080e-02 -3.34071703e-02  4.83248569e-02 -6.16907887e-02  4.52203304e-02  3.44095728e-03 -2.51690317e-02  3.40977162e-02  2.46876478e-02  4.67401510e-03  2.12683938e-02 -3.45606962e-03  1.80950016e-02
 -6.65076971e-02 -5.68277426e-02 -6.88709766e-02 -1.30293500e-02 -2.37979759e-02  1.80553887e-02 -7.72333667e-02 -8.56943522e-03  1.91549510e-02 -8.33140779e-03 -1.34733866e-03 -4.83256094e-02 -2.80473218e-03
 -2.44241953e-03 -9.87402499e-02 -8.24460983e-02  4.14892398e-02 -1.30495839e-02  4.34679799e-02  4.49994812e-03  4.42992002e-02 -9.13854316e-02  8.43153708e-03  1.68169960e-02 -3.91042270e-02 -7.98141491e-03
 -5.16823232e-02 -5.08615039e-02 -4.53437679e-02 -4.98923734e-02 -4.51540537e-02 -8.56273845e-02  2.09337529e-02 -5.55219464e-02  1.76757749e-03 -6.13401411e-03  3.62911820e-02 -2.59846523e-02 -2.00108588e-02
 -3.93580273e-02  5.06016007e-03 -5.37365526e-02  3.22223566e-02 -2.28474010e-03  1.29975844e-02 -6.12050407e-02 -1.05224131e-03 -1.58257608e-03 -4.14088927e-02 -1.02844052e-02 -5.23678958e-02  2.69939993e-02
 -3.45915966e-02 -6.54063895e-02 -4.62423712e-02  7.93707818e-02  2.52677985e-02 -2.93134525e-02  1.00937970e-02  1.18398434e-02 -3.78319249e-02 -1.68423932e-02  1.76228676e-02 -4.91100065e-02  1.19618420e-02
  2.70030536e-02 -4.68529761e-02 -7.50001520e-03 -8.01536813e-03  3.65575328e-02  8.50174855e-03 -6.35103509e-02 -4.30282131e-02  1.80962496e-02  3.02586183e-02  1.34778740e-02 -2.09567212e-02 -4.30926085e-02
 -1.57399476e-02  4.75276969e-02 -1.04921788e-01  1.97727457e-02  9.73437447e-03 -8.33401456e-02 -5.27075492e-03 -3.57254036e-02 -5.51621020e-02 -6.66137189e-02  4.46118452e-02 -4.73510213e-02 -3.30080092e-02
  2.40020384e-03  4.54113074e-02  2.43031066e-02  5.40856533e-02  2.49560121e-02  5.62681956e-03 -1.61635261e-02 -2.08627470e-02 -1.91719607e-02 -1.15571348e-02  5.26386686e-03  1.51769742e-02  9.21672303e-03
  6.13655243e-03 -6.55378923e-02 -1.06840506e-01 -3.98056302e-03 -1.94518305e-02  4.10661846e-02  1.47867743e-02 -1.11436406e-02 -1.74489878e-02 -3.43100950e-02 -1.93871036e-02  5.50077148e-02 -8.73839557e-02
  2.57645790e-02 -2.24715509e-02 -2.68510226e-02  3.48799350e-03 -1.11021725e-02  5.26228882e-02  1.36931678e-02 -4.72956598e-02 -8.41976628e-02 -2.92517208e-02 -2.97983806e-03 -1.45290848e-02 -2.76376065e-02
 -3.11349705e-02  1.94145609e-02  2.06362270e-02 -7.34852925e-02 -3.05439550e-02 -9.04708821e-03 -3.20820101e-02 -4.73959168e-04 -5.42684225e-04 -1.56734306e-02 -7.62691768e-03 -3.17123123e-02  2.05654632e-02
 -1.92003399e-02 -2.94904672e-02 -9.03217793e-02 -4.98675741e-02 -1.08533194e-02 -5.24595343e-02 -9.39057767e-02 -5.83649203e-02 -2.14326382e-02 -2.36656796e-03 -3.76531556e-02 -8.00841227e-02  4.35419641e-02
 -8.84294335e-04 -4.35534306e-02 -4.52498868e-02 -4.06502821e-02 -6.92172274e-02  4.17556688e-02 -6.05128184e-02 -6.48240140e-03 -3.72381322e-02 -5.55567793e-04 -6.58582598e-02  9.89968982e-03 -5.14916219e-02
 -3.48042836e-03 -1.83021147e-02 -2.60708760e-03 -7.42479861e-02 -4.47370633e-02 -4.32397425e-02 -2.53343992e-02 -5.27679175e-03 -3.93298902e-02  1.66524332e-02  2.71635558e-02 -6.23826915e-03 -9.62183462e-04
 -1.62978563e-02 -6.61583468e-02 -2.20692698e-02 -6.61713034e-02  4.64543775e-02 -2.31184084e-02  1.21555058e-02  1.71498675e-02 -3.32990326e-02  6.40662527e-03 -2.23796982e-02  2.51753032e-02 -2.59028189e-02
  2.62370948e-02  1.65636912e-02 -2.83479467e-02  3.74291725e-02 -6.68942854e-02 -3.25272605e-02  1.19628031e-02 -3.39038335e-02  1.57797467e-02  1.69692691e-02 -2.36886926e-02 -3.91316004e-02 -3.23203020e-02
  1.37892263e-02 -3.87814008e-02 -1.02040432e-02 -2.86921561e-02 -4.69354615e-02 -1.57062302e-03  4.55904007e-03  2.53503118e-02 -2.94023799e-03  7.95436837e-03  4.70472760e-02 -4.03328380e-03  2.48316564e-02
 -1.53100602e-02 -2.70378459e-02 -2.13968661e-03  1.00920834e-02  8.08873177e-02  2.31958739e-02  2.06584651e-02  1.78932343e-02 -4.16267999e-02  4.10719849e-02 -2.30729841e-02 -2.64290571e-02 -6.63546994e-02
  4.01059911e-02 -9.29727219e-03 -7.50036016e-02 -4.15391251e-02  1.37821184e-02 -5.01939915e-02 -5.76973297e-02 -7.65870279e-03 -5.35742901e-02  3.53098381e-03 -8.10436308e-02  2.24992428e-02 -6.38229102e-02
 -4.35542390e-02 -6.99904002e-03  4.76081017e-03 -1.39805849e-03  1.17315678e-02 -5.72536923e-02  2.18869094e-02  1.77835207e-02 -2.85619550e-04  2.63929572e-02 -1.12051526e-02  1.85196809e-02  1.38412677e-02
  1.18629681e-02]
```


Model	Hidden Size (Embedding Dim) d
BERT-base	768
BERT-large	1024
DistilBERT	768
RoBERTa-base	768
RoBERTa-large	1024
GPT-2 (small)	768
GPT-2 (medium)	1024
GPT-2 (large)	1280
GPT-2 (XL)	1600
GPT-3 (Davinci)	12288



https://www.datacamp.com/tutorial/how-transformers-work

How are **Embeddings** created? (via transformers)

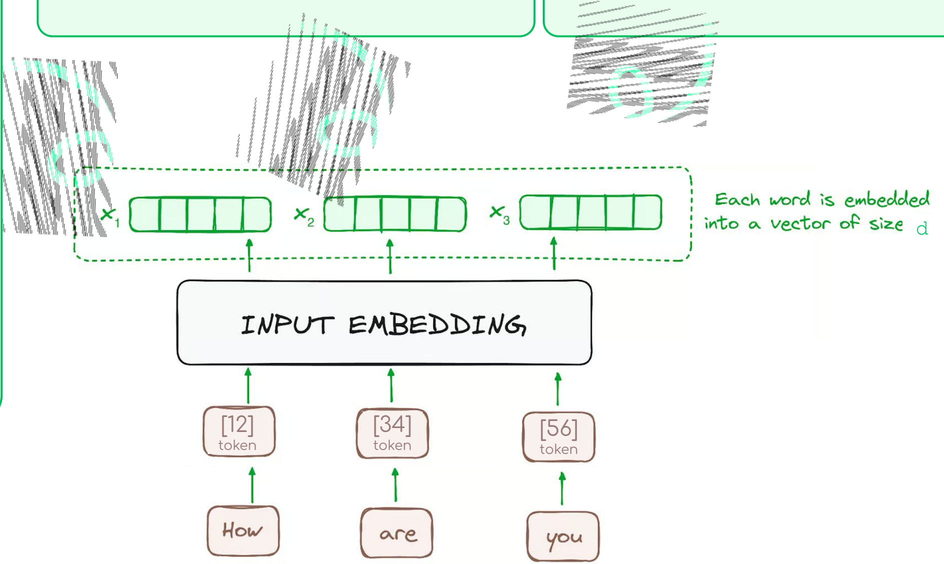
📌 Word: How
🧩 Tokenized: ['how']
🔢 Token ID: 2129

📌 Word: are
🧩 Tokenized: ['are']
🔢 Token ID: 2024
🔷 Static Embedding Vector (shape: (768,)):

📌 Word: you
🧩 Tokenized: ['you']
🔢 Token ID: 2017
🔷 Static Embedding Vector (shape: (768,)):

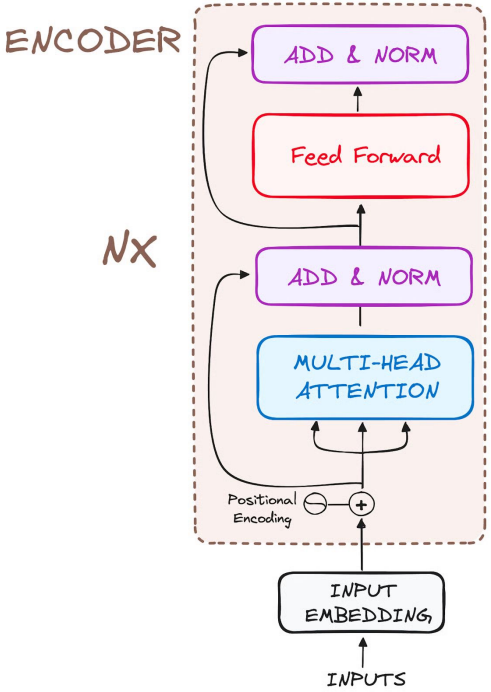
🔷 Static Embedding Vector for 'How' (shape: (768,)):

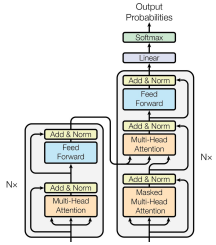
[-2.40632426e-02 3.17239240e-02 -7.56581724e-02 6.26911037e-03 -2.58011918e-04 -4.89562452e-02 -8.62864181e-02 -1.48444436e-02 2.95870882e-02 -2.93501709e-02 -1.86109450e-02 -2.58933883e-02 4.38467599e-02
5.06787337e-02 -2.69273855e-02 -9.60835367e-02 2.47513223e-02 -1.66852530e-02 4.85536605e-02 1.61708891e-02 -1.12872068e-02 2.37299539e-02 -8.33070744e-03 -3.69310901e-02 -2.43126173e-02 -5.99710383e-02
-1.38285588e-02 -7.60446936e-02 7.07411915e-02 -2.73569897e-02 -3.67836654e-02 -7.62912259e-02 -8.92018452e-02 6.45733774e-02 -3.21142282e-03 -3.13841812e-02 -7.32790530e-02 -2.19877567e-02 -4.54654545e-03
-5.16635329e-02 -8.74689315e-03 1.45732490e-02 -8.29743370e-02 6.72775134e-02 3.09977941e-02 4.56132321e-03 2.29086820e-02 -3.61224562e-02 3.17416489e-02 1.83614213e-02 -4.02215775e-03 -4.04329039e-02
3.96960974e-03 5.34170587e-03 -1.11106066e-02 9.45139024e-03 -4.14175689e-02 1.76874711e-03 -1.97757743e-02 -2.28614025e-02 -9.49357823e-03 8.20351951e-03 -1.82070285e-02 -3.47239673e-02 4.16595377e-02
-5.05649485e-02 -8.35249200e-03 1.25095462e-02 4.21202416e-03 -2.61297952e-02 -7.05806613e-02 -2.02179626e-02 -3.15761529e-02 -4.31864224e-02 3.85294226e-03 7.09294435e-03 -3.02653443e-02 1.60581004e-02
4.92024329e-03 -2.77611353e-02 3.49431969e-02 2.80356519e-02 3.50493006e-02 2.73176637e-02 -1.19978096e-02 -8.49911273e-02 -4.09458280e-02 -2.72356793e-02 -7.49194324e-02 -7.59286061e-02 4.99165803e-02
-4.78370413e-02 -6.51342273e-02 -3.75630707e-02 -3.04567311e-02 -6.00833967e-02 -5.35118720e-03 -1.03650177e-02 -3.19029652e-02 -3.16432454e-02 4.52513341e-03 6.43489733e-02 -8.88413377e-03 -9.86471213e-03
-1.52054569e-02 -2.22128467e-03 8.31383560e-03 -2.57547889e-02 -1.01689184e-02 -8.10694299e-04 -6.42709807e-02 -4.49671820e-02 -3.81466933e-02 -4.20909375e-02 -4.54924069e-02 7.92594850e-02 8.85786768e-03
1.45143382e-02 -1.64769571e-02 -1.84432361e-02 -7.20525673e-03 2.29655154e-04 1.04185855e-02 1.79463569e-02 8.97494785e-04 -5.32222316e-02 -1.26120383e-02 -5.76113574e-02 -1.71773471e-02 2.03900337e-02
-1.71129815e-02 8.24113283e-03 2.27314793e-03 6.85477406e-02 -6.80906326e-02 3.66026238e-02 3.42215300e-02 2.34122779e-02 -4.25386056e-02 -6.82519702e-03 3.85989621e-02 7.46158464e-03 3.85151245e-02
-3.69149111e-02 3.06216143e-02 -7.47312456e-02 -1.02839796e-02 -3.08722034e-02 7.31505966e-03 -8.44044462e-02 4.60562184e-02 2.53819246e-02 -5.24426326e-02 -2.34480351e-02 -2.61210948e-02 -2.73317080e-02
2.94399988e-02 -2.03207750e-02 -2.91144606e-02 2.75079440e-02 -5.10416701e-02 5.68141742e-03 -3.95116992e-02 9.67309996e-03 4.10742068e-05 -7.00547323e-02 -5.13092540e-02 -5.24353981e-02 1.59527287e-02
4.54671569e-02 9.29208938e-04 -6.32209256e-02 -3.60561609e-02 1.03921546e-02 -3.00257392e-02 -1.52557818e-02 8.80255736e-03 5.90848504e-03 -3.73701751e-02 -1.89466793e-02 -4.74576699e-03 -5.00825047e-02
-1.03363851e-02 -3.23428623e-02 1.14934720e-01 2.80265342e-02 -6.46438356e-03 -7.34714642e-02 8.06048587e-02 -4.89006229e-02 2.72423513e-02 2.29129428e-03 -5.01409620e-02 -5.81299774e-02 -1.98586285e-02
-2.16328800e-02 -2.25455519e-02 -1.76306274e-02 2.31720675e-02 3.54568698e-02 -1.79572664e-02 2.39172839e-02 -1.86421890e-02 -7.19889179e-02 -1.86346546e-02 3.24567296e-02 -5.17822616e-02 1.19499033e-02
2.79369392e-02 -6.09460287e-02 -4.28852178e-02 2.25416645e-02 -2.86062695e-02 -3.66943628e-02 3.00906040e-02 -3.10584940e-02 -6.72834590e-02 -4.79315780e-02 3.69124045e-03 -2.50349734e-02 4.02781442e-02
7.43498141e-03 -5.60195372e-02 -1.35053573e-02 -3.45139615e-02 3.79546955e-02 5.51038310e-02 -3.35163772e-02 5.03432900e-02 -9.50830814e-04 -2.85925046e-02 1.77440383e-02 1.03203962e-02 -1.42628606e-02
1.98729225e-02 4.85438630e-02 4.03669588e-02 6.81296410e-03 -8.47603157e-02 1.96953397e-02 6.08333526e-03 2.46805009e-02 -3.51215117e-02 -3.49530317e-02 -1.43865198e-02 3.60011533e-02 -2.97866575e-02
-2.72921775e-03 -1.59283988e-02 -9.03266203e-03 -4.00798805e-02 -3.95079665e-02 1.17932884e-02 -2.03591753e-02 1.05042495e-02 -2.03903019e-02 -1.80941001e-02 -3.59604321e-02 -9.18738842e-02 6.09645434e-02
1.80536543e-03 2.56685866e-03 -3.29396017e-02 1.34370727e-02 1.21362126e-02 -3.67373712e-02 -8.94654635e-03 -4.06659022e-02 2.78272592e-02 9.23155714e-03 1.16159860e-02 -4.49797474e-02 1.41576407e-02
-1.11131638e-01 1.03015658e-02 -4.55153082e-03 -1.28376996e-02 -7.35259429e-02 2.52520982e-02 -2.76803765e-02 -3.90702896e-02 -3.61095928e-02 -4.60212603e-02 1.90247912e-02 4.47232239e-02 -1.42136076e-02
1.01556890e-02 2.36594882e-02 -4.32280488e-02 1.24429623e-02 -4.39441130e-02 -1.82906911e-02 4.15928699e-02 -6.13628216e-02 4.86430107e-03 2.65281480e-02 -3.06413067e-03 -4.58845543e-03 1.36416610e-02
-5.12329116e-02 4.49450947e-02 -3.06420941e-02 3.11115067e-02 5.19374460e-02 -3.73131372e-02 -4.36187163e-02 1.31423762e-02 -7.99829606e-03 -6.31169006e-02 -2.69059557e-02 -1.93704180e-02 -2.04635691e-03
8.94358009e-03 -6.86627105e-02 -6.04309589e-02 2.33461652e-02 -7.06690596e-04 -4.44739945e-02 1.83669981e-02 -1.50816711e-02 -5.70608526e-02 -6.09482639e-04 -2.96346750e-02 1.66279897e-02 4.40274552e-02
-7.93768093e-03 2.15591900e-02 -8.25689435e-02 1.33756967e-02 -4.13693562e-02 -4.56367712e-03 -7.26461969e-03 2.36548739e-03 -1.13174543e-01 -8.58894959e-02 -3.93058993e-02 5.97350579e-03 -3.45391035e-02
3.41741815e-02 -9.25091878e-02 -5.73617369e-02 -3.12605239e-02 -8.67043156e-03 -5.94054768e-03 1.92349888e-02 9.78673249e-03 7.47569501e-02 -3.05492338e-02 -1.03461161e-01 5.84605336e-02 1.74750816e-02
-5.44028766e-02 4.93369475e-02 -1.96900498e-02 9.62559320e-03 -4.25645970e-02 -8.98296013e-02 -5.03054932e-02 -5.04233837e-02 -6.15962669e-02 -3.69961634e-02 -2.78792251e-02 8.13827757e-03 3.73848947e-03
5.26996851e-02 1.62710864e-02 5.72457677e-04 -3.81550156e-02 -4.81124362e-03 -9.95321944e-02 1.06638689e-02 -3.81066315e-02 -2.29458977e-02 -6.60021529e-02 3.84338684e-02 -3.70474532e-02 -1.22215413e-03
6.31565526e-02 1.39290588e-02 2.92073451e-02 3.00167818e-02 -7.62918293e-02 -2.23823953e-02 -6.00151010e-02 3.99245955e-02 -5.81338024e-03 -1.29170343e-02 2.33683120e-02 -3.22067700e-02 -2.15006135e-02
-7.66447410e-02 -3.54666859e-02 -9.19029675e-03 -2.27978323e-02 -2.17447542e-02 -4.49042954e-02 9.99037363e-03 -2.96426043e-02 -2.87224278e-02 -2.63543427e-02 -3.99767496e-02 -2.76514534e-02 -4.02079299e-02
-6.59588277e-02 6.84235394e-02 -8.39904472e-02 -2.62792427e-02 1.81724560e-02 -1.36873908e-02 -4.11824621e-02 -3.11036073e-02 -7.22710509e-03 -4.45440374e-02 -7.06092715e-02 -1.50430547e-02 -9.25701484e-03
-3.85007672e-02 -1.26395272e-02 3.74264009e-02 4.08810377e-02 2.48979535e-02 -4.75944355e-02 -2.07229517e-02 1.01020150e-02 -4.31836694e-02 -3.27098407e-02 2.57209986e-02 -8.33619386e-03 -2.78881397e-02
2.67299212e-04 3.83807234e-02 -2.68067524e-04 -3.54761295e-02 -4.68954369e-02 -2.21075229e-02 1.88480578e-02 -6.35658652e-02 -2.07822993e-02 -3.00569627e-02 -2.63915136e-02 -3.51044759e-02 -2.96256281e-02
-4.72513363e-02 -8.74045957e-03 -5.50780329e-04 -6.05693720e-02 -6.11014515e-02 1.91002674e-02 2.26630848e-02 3.75893265e-02 -2.15478893e-02 -7.14205857e-03 1.22040827e-02 3.28841782e-03 -3.31071988e-02
1.28081040e-02 3.57418172e-02 -1.59375686e-02 -1.41410902e-03 4.88664806e-02 -9.71987993e-02 -2.06610584e-03 -1.89623013e-02 9.96184722e-03 4.53704782e-02 1.96120664e-02 -1.68459173e-02 1.30380057e-02
-1.66016687e-02 3.19261774e-02 8.40354525e-03 2.03369930e-03 -8.23270809e-03 -1.20725464e-02 -5.65328747e-02 -1.67112183e-02 -5.16212694e-02 4.16195579e-02 4.16581705e-03 -2.00486090e-03 -4.85344604e-02
-1.62212793e-02 -6.42460072e-03 3.35895419e-02 2.81633604e-02 2.43080556e-02 2.69450247e-02 1.66514739e-02 -1.75541081e-02 3.39950845e-02 -5.51930964e-02 -3.53590287e-02 -3.94096710e-02 -6.34305254e-02
-3.63813080e-02 -3.34071703e-02 4.83248569e-02 -6.16907887e-02 4.52203304e-02 3.44095728e-03 -2.51690317e-02 3.40977162e-02 2.46876478e-02 4.67401510e-03 2.12683938e-02 -3.45606962e-03 1.80950016e-02
-6.65076971e-02 -5.68277426e-02 -6.88709766e-02 -1.30293500e-02 -2.37979759e-02 1.80553887e-02 -7.72333667e-02 -8.56943522e-03 1.91549510e-02 -8.33140779e-03 -1.34733866e-03 -4.83256094e-02 -2.80473218e-03
-2.44241953e-03 -9.87402499e-02 -8.24460983e-02 4.14892398e-02 -1.30495839e-02 4.34679799e-02 4.49994812e-03 4.42992002e-02 -9.13854316e-02 8.43153708e-03 1.68169960e-02 -3.91042270e-02 -7.98141491e-03
-5.16823232e-02 -5.08615039e-02 -4.53437679e-02 -4.98923734e-02 -4.51540537e-02 -8.56273845e-02 2.09337529e-02 -5.55219464e-02 1.76757749e-03 -6.13401411e-03 3.62911820e-02 -2.59846523e-02 -2.00108588e-02
-3.93580273e-02 5.06016007e-03 -5.37365526e-02 3.22223566e-02 -2.28474010e-03 1.29975844e-02 -6.12050407e-02 -1.05224131e-03 -1.58257608e-03 -4.14088927e-02 -1.02844052e-02 -5.23678958e-02 2.69939993e-02
-3.45915966e-02 -6.54063895e-02 -4.62423712e-02 7.93707818e-02 2.52677985e-02 -2.93134525e-02 1.00937970e-02 1.18398434e-02 -3.78319249e-02 -1.68423932e-02 1.76228676e-02 -4.91100065e-02 1.19618420e-02
2.70030536e-02 -4.68529761e-02 -7.50001520e-03 -8.01536813e-03 3.65575328e-02 8.50174855e-03 -6.35103509e-02 -4.30282131e-02 1.80962496e-02 3.02586183e-02 1.34778740e-02 -2.09567212e-02 -4.30926085e-02
-1.57399476e-02 4.75276969e-02 -1.04921788e-01 1.97727457e-02 9.73437447e-03 -8.33401456e-02 -5.27075492e-03 -3.57254036e-02 -5.51621020e-02 -6.66137189e-02 4.46118452e-02 -4.73510213e-02 -3.30080092e-02
2.40020384e-03 4.54113074e-02 2.43031066e-02 5.40856533e-02 2.49560121e-02 5.62681956e-03 -1.61635261e-02 -2.08627470e-02 -1.91719607e-02 -1.15571348e-02 5.26386686e-03 1.51769742e-02 9.21672303e-03
6.13655243e-03 -6.55378923e-02 -1.06840506e-01 -3.98056302e-03 -1.94518305e-02 4.10661846e-02 1.47867743e-02 -1.11436406e-02 -1.74489878e-02 -3.43100950e-02 -1.93871036e-02 5.50077148e-02 -8.73839557e-02
2.57645790e-02 -2.24715509e-02 -2.68510226e-02 3.48799350e-03 -1.11021725e-02 5.26228882e-02 1.36931678e-02 -4.72956598e-02 -8.41976628e-02 -2.92517208e-02 -2.97983806e-03 -1.45290848e-02 -2.76376065e-02
-3.11349705e-02 1.94145609e-02 2.06362270e-02 -7.34852925e-02 -3.05439550e-02 -9.04708821e-03 -3.20820101e-02 -4.73959168e-04 -5.42684225e-04 -1.56734306e-02 -7.62691768e-03 -3.17123123e-02 2.05654632e-02
-1.92003399e-02 -2.94904672e-02 -9.03217793e-02 -4.98675741e-02 -1.08533194e-02 -5.24595343e-02 -9.39057767e-02 -5.83649203e-02 -2.14326382e-02 -2.36656796e-03 -3.76531556e-02 -8.00841227e-02 4.35419641e-02
-8.84294335e-04 -4.35534306e-02 -4.52498868e-02 -4.06502821e-02 -6.92172274e-02 4.17556688e-02 -6.05128184e-02 -6.48240140e-03 -3.72381322e-02 -5.55567793e-04 -6.58582598e-02 9.89968982e-03 -5.14916219e-02
-3.48042836e-03 -1.83021147e-02 -2.60708760e-03 -7.42479861e-02 -4.47370633e-02 -4.32397425e-02 -2.53343992e-02 -5.27679175e-03 -3.93298902e-02 1.66524332e-02 2.71635558e-02 -6.23826915e-03 -9.62183462e-04
-1.62978563e-02 -6.61583468e-02 -2.20692698e-02 -6.61713034e-02 4.64543775e-02 -2.31184084e-02 1.21555058e-02 1.71498675e-02 -3.32990326e-02 6.40662527e-03 -2.23796982e-02 2.51753032e-02 -2.59028189e-02
2.62370948e-02 1.65636912e-02 -2.83479467e-02 3.74291725e-02 -6.68942854e-02 -3.25272605e-02 1.19628031e-02 -3.39038335e-02 1.57797467e-02 1.69692691e-02 -2.36886926e-02 -3.91316004e-02 -3.23203020e-02
1.37892263e-02 -3.87814008e-02 -1.02040432e-02 -2.86921561e-02 -4.69354615e-02 -1.57062302e-03 4.55904007e-03 2.53503118e-02 -2.94023799e-03 7.95436837e-03 4.70472760e-02 -4.03328380e-03 2.48316564e-02
-1.53100602e-02 -2.70378459e-02 -2.13968661e-03 1.00920834e-02 8.08873177e-02 2.31958739e-02 2.06584651e-02 1.78932343e-02 -4.16267999e-02 4.10719849e-02 -2.30729841e-02 -2.64290571e-02 -6.63546994e-02
4.01059911e-02 -9.29727219e-03 -7.50036016e-02 -4.15391251e-02 1.37821184e-02 -5.01939915e-02 -5.76973297e-02 -7.65870279e-03 -5.35742901e-02 3.53098381e-03 -8.10436308e-02 2.24992428e-02 -6.38229102e-02
-4.35542390e-02 -6.99904002e-03 4.76081017e-03 -1.39805849e-03 1.17315678e-02 -5.72536923e-02 2.18869094e-02 1.77835207e-02 -2.85619550e-04 2.63929572e-02 -1.12051526e-02 1.85196809e-02 1.38412677e-02
1.18629681e-02]



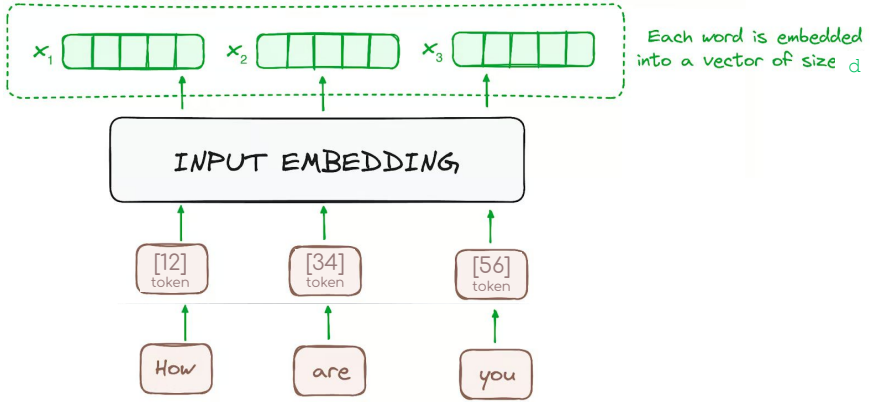
https://www.datacamp.com/tutorial/how-transformers-work

How are **Embeddings** created? (via transformers)



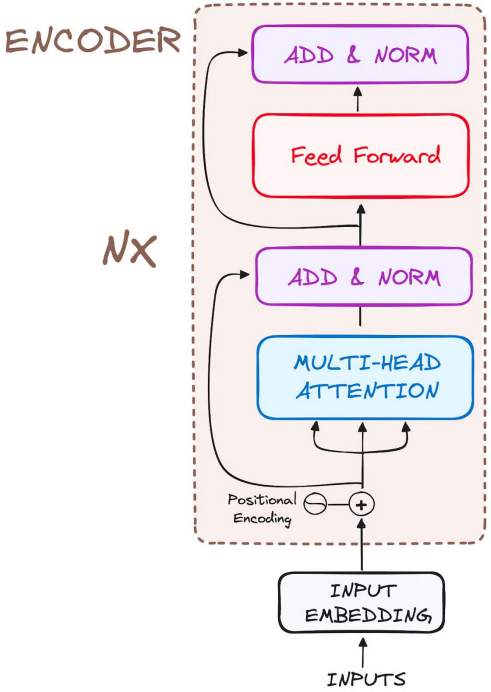


STATIC EMBEDDING.

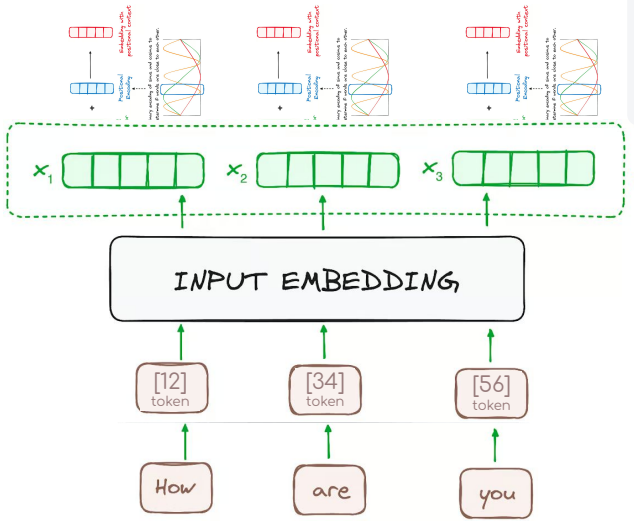


https://www.datacamp.com/tutorial/how-transformers-work

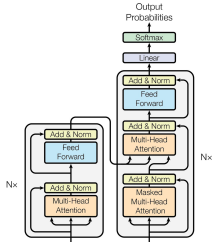
How are **Embeddings** created? (via transformers)



STATIC EMBEDDING.



Each word is embedded into a vector of size d

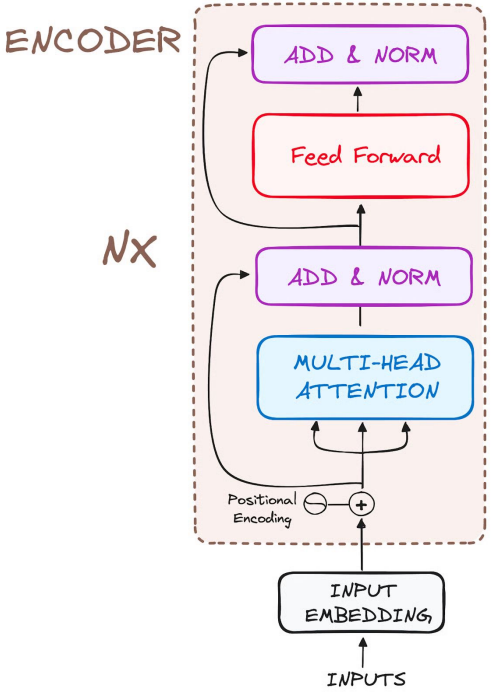


The **positional encoding** adds a "where it is" signal

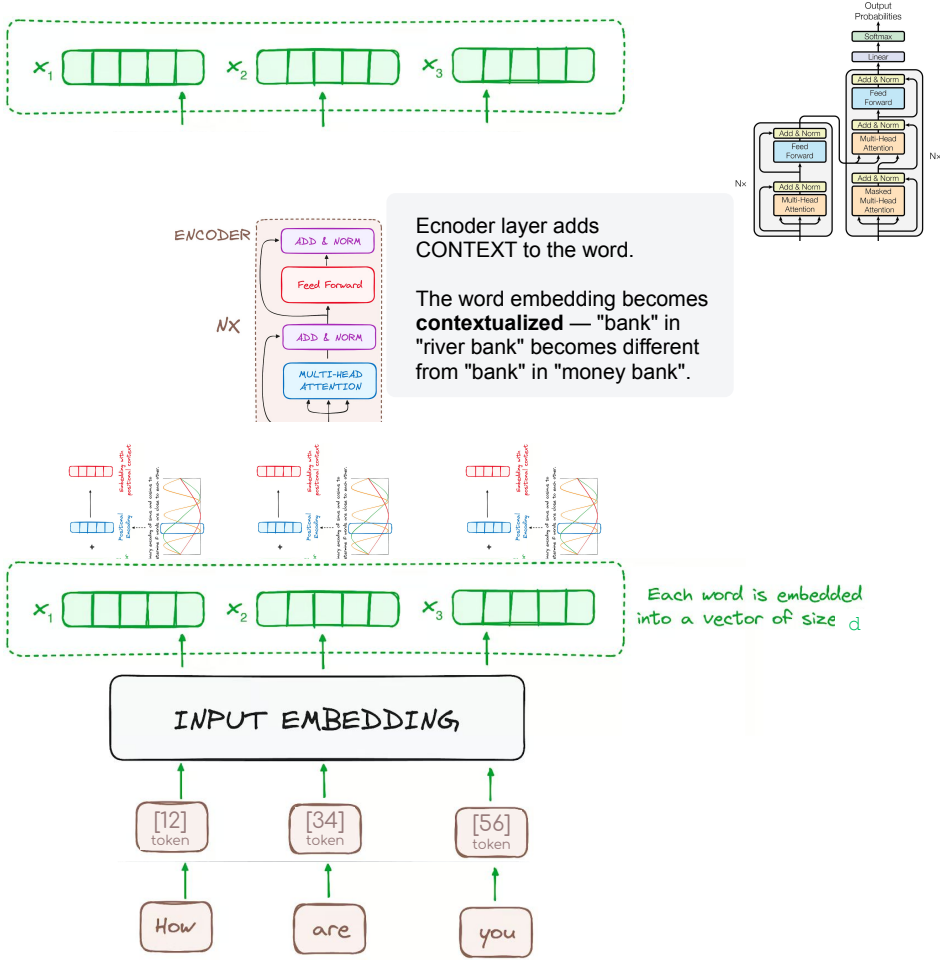
Two identical words at different positions get **different input vectors** because their embeddings are offset by different positional encodings

https://www.datacamp.com/tutorial/how-transformers-work

How are **Embeddings** created? (via transformers)

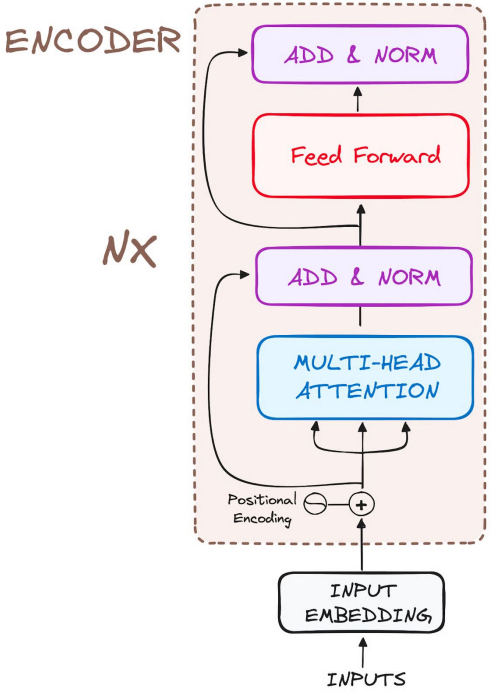


STATIC EMBEDDING.

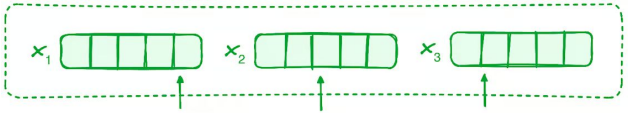


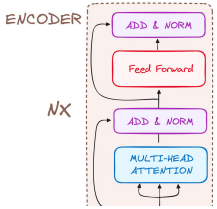
https://www.datacamp.com/tutorial/how-transformers-work

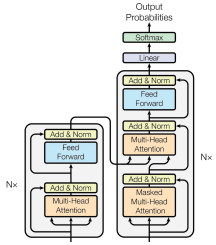
How are **Embeddings** created? (via transformers)



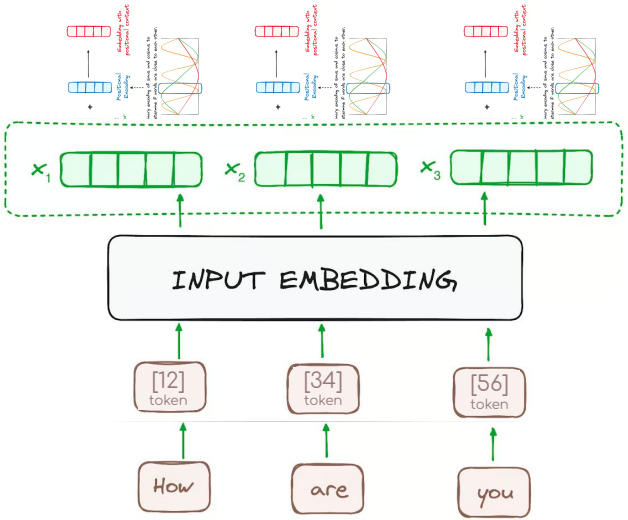
CONTEXTUAL EMBEDDING.







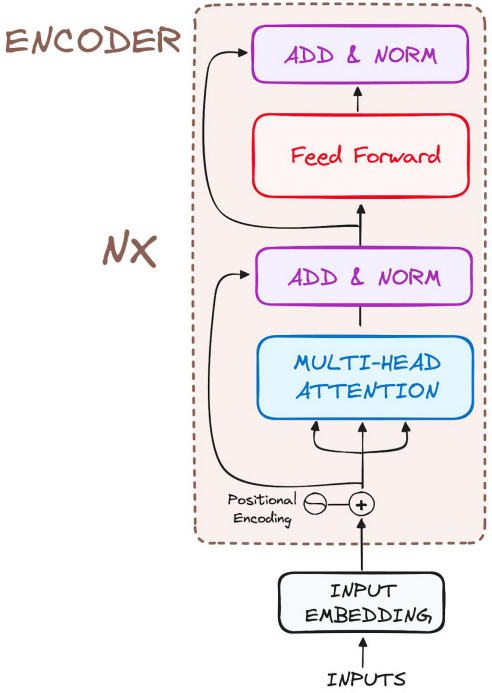
STATIC EMBEDDING.



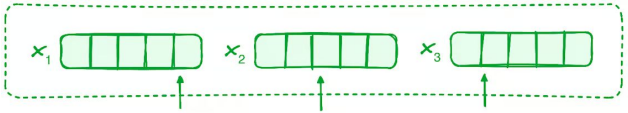
Each word is embedded into a vector of size d

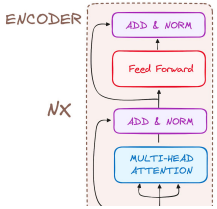
https://www.datacamp.com/tutorial/how-transformers-work

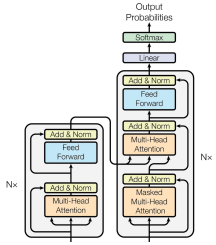
How are **Embeddings** created? (via transformers)



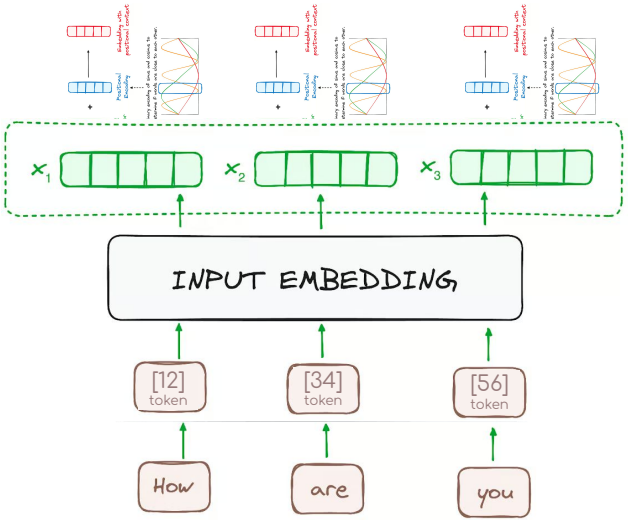
CONTEXTUAL EMBEDDING.







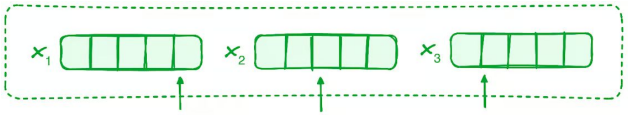
STATIC EMBEDDING.



Each word is embedded into a vector of size d

https://www.datacamp.com/tutorial/how-transformers-work

How are **Embeddings** created? (via transformers)



CONTEXTUAL EMBEDDING.

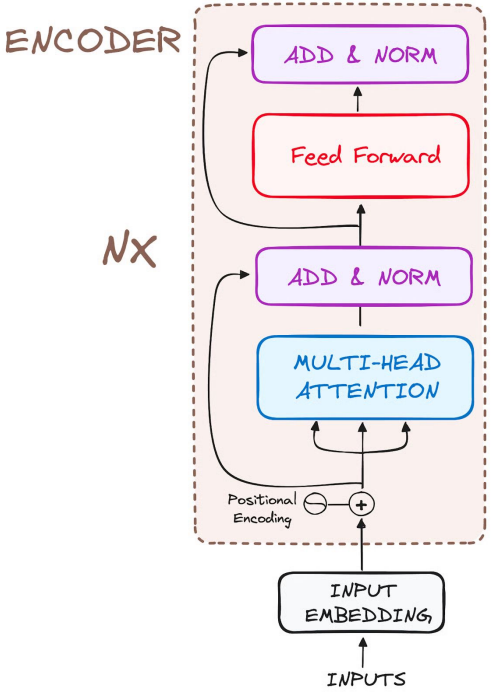
```
================================================================================
🔍 Comparison of Static vs Contextual Embeddings for the Word 'bank'

📌 Input Sentence: He sat by the river bank.
🔢 'bank' Token ID: 2924
🧊 Static Embedding (first 5 dims): [-0.019130440428853035, -0.0645589530467987, -0.09128748625516891, -0.07761748880147934, -0.025318529456853867]
🔥 Contextual Embedding (first 5 dims): [0.1599486917257309, -0.3381432890892029, -0.03246789425611496, -0.08658606559038162, -0.39891570806503296]
--------------------------------------------------------------------------------
📌 Input Sentence: She deposited money at the bank.
🔢 'bank' Token ID: 2924
🧊 Static Embedding (first 5 dims): [-0.019130440428853035, -0.0645589530467987, -0.09128748625516891, -0.07761748880147934, -0.025318529456853867]
🔥 Contextual Embedding (first 5 dims): [0.393318772315979, -0.4198230504989624, -0.2934713363647461, 0.019461048766970634, 0.853203296661377]
--------------------------------------------------------------------------------
```

Note: Static vs Contextual Embeddings

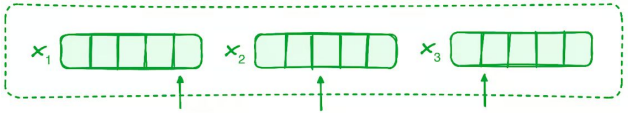
In both sentences, the word **"bank"** has the **same static embedding** because it's based only on the token itself.
But its **contextual embedding differs** — because BERT adjusts the vector based on surrounding words.
This shows how Transformers understand meaning based on **context**, not just the word.

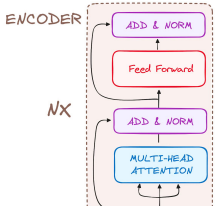
https://www.datacamp.com/tutorial/how-transformers-work

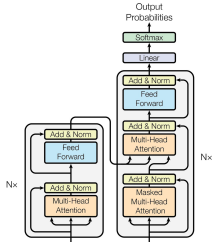
How are **Embeddings** created? (via transformers)



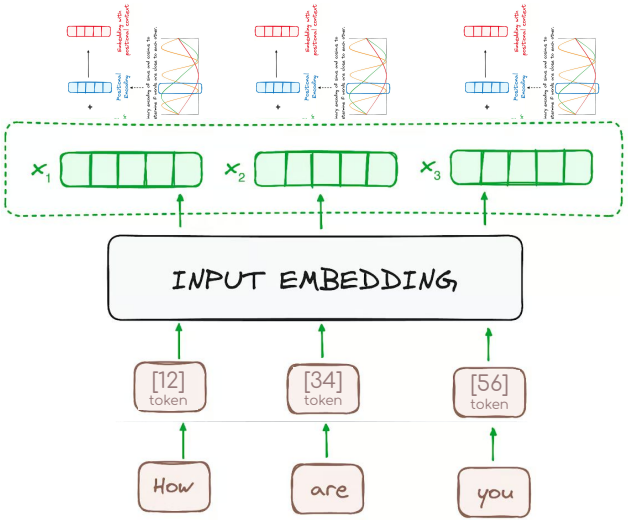
CONTEXTUAL EMBEDDING.







STATIC EMBEDDING.

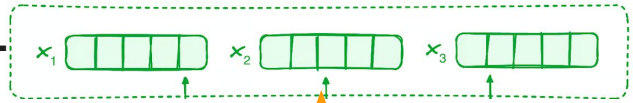


Each word is embedded into a vector of size d

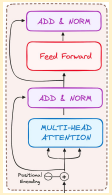
https://www.datacamp.com/tutorial/how-transformers-work

All **Embedding Models** are Encoders under the hood

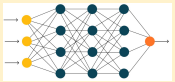
CONTEXTUAL EMBEDDING.

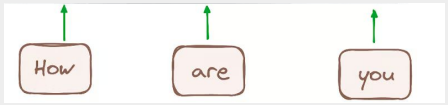


Vectors



EMBEDDING MODEL





Words

All **Embedding Models** are Encoders under the hood

OPEN-SOURCE

LLM

- **LLaMA 2** (by Meta)
- **OpenLLAMA** (by Berkeley AI)
- **Gemma** (by Google)
- **Mixtral** (by Mistral AI)
- **Mistral** (by Mistral AI)
- **Dolly** (by Databricks)
- **Falcon** (by TII)
- **DeepSeek** (by DeepSeek AI)

Vision

- **SAM** (by Meta AI)
- **Detectron2** (by Facebook AI R)
- **MMDetection** (by OpenMMlab)
- **YOLOv8** (by Ultralytics)

VLM

- **MiniGPT-4**
- **BLIP-2** (by Anthropic)
- **BLIP** (by Salesforce)
- **CLIP** (by Open AI)

Audio

- **ESPnet**
- **Wav2Vec**
- **Whisper**

Word2Vec

- **Word2Vec** (by PyTorch)
- **Word2Vec** (by Gensim)
- **FastTeext** (by Facebook)
- **SentenceTransformer** (by Huggingface)

CLOSED-SOURCE

LLM

- **GPT-4** (by OpenAI)
- **Claude 3** (by Anthropic)
- **Gemini** (by Google DeepMind)
- **Command R+** (by Cohere)
- **Ernie Bot** (by Baidu)
- **Qwen** (by Alibaba)
- **SenseChat** (by SenseTime)
- **Grok** (by xAI)

Vision

- **Amazon Rekognition** (by AWS)

VLM

- **GPT-4V** (by OpenAI)
- **Claude 4 Opus** (by Anthropic)
- **Gemini** (by Google DeepMind)
- **Grok Vision** (by xAI)
- **Imagen**
- **Midjourney**
- **Flux**

Audio

- **ElevenLabs**
- **Google Speech-to-Text**
- **Amazon Transcribe**
- **Microsoft Azure Speech**
- **Descript Overdub**

Word2Vec

- **OpenAI Embeddings**
- **Google Cloud Embeddings**
- **Cohere Embeddings**
- **Amazon Comprehend**

LLM vs Word2Vec (Embedding Models)

OPEN-SOURCE

- **LLaMA 2** (by Meta)
- **OpenLLAMA** (by Berkeley AI)
- **Gemma** (by Google)
- **Mixtral** (by Mistral AI)
- **Mistral** (by Mistral AI)
- **Dolly** (by Databricks)
- **Falcon** (by TII)
- **DeepSeek** (by DeepSeek AI)

LLM

CLOSED-SOURCE

- **GPT-4** (by OpenAI)
- **Claude 3** (by Anthropic)
- **Gemini** (by Google DeepMind)
- **Command R+** (by Cohere)
- **Ernie Bot** (by Baidu)
- **Qwen** (by Alibaba)
- **SenseChat** (by SenseTime)
- **Grok** (by xAI)

LLM

- Input: sequence of words
- Output: sequence of words

Word2Vec (Embedding Models)

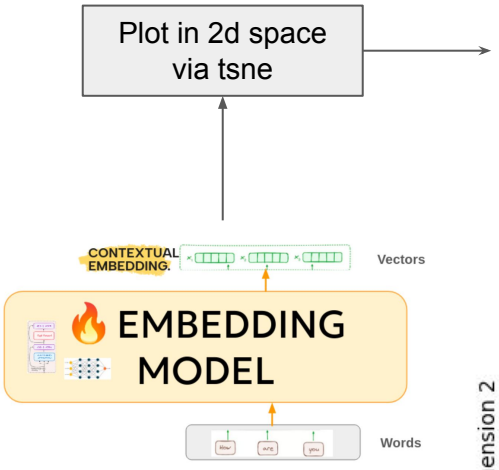
- Input: word / sentences
- Output: vectors

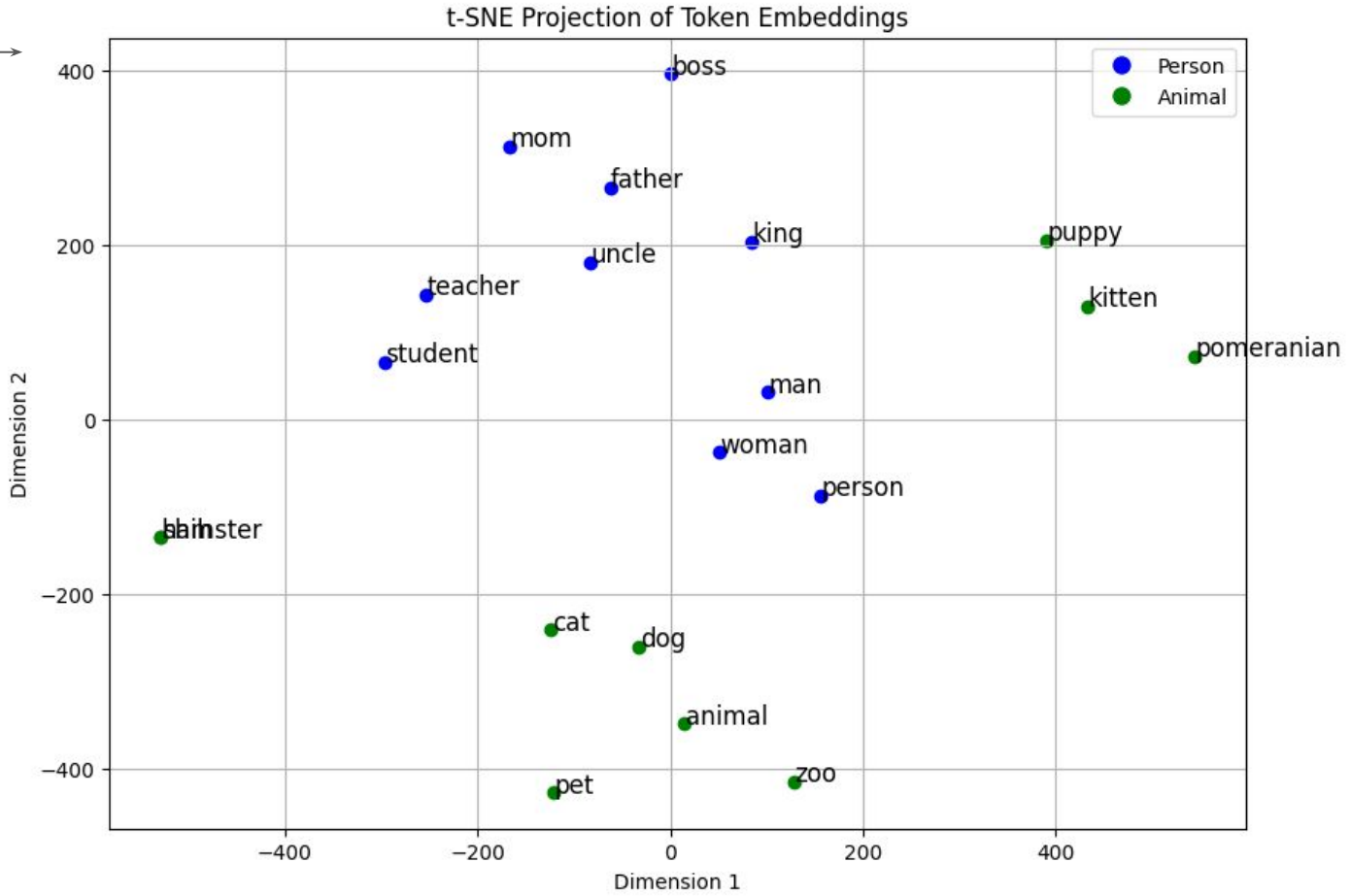
- **Word2Vec** (by PyTorch)
- **Word2Vec** (by Gensim)
- **FastTeext** (by Facebook)
- **SentenceTransformer** (by Huggingface)

Word2Vec

- **OpenAI Embeddings**
- **Google Cloud Embeddings**
- **Cohere Embeddings**
- **Amazon Comprehend**

From context embedding (in 2d space)





From context embedding to **sentence embedding**

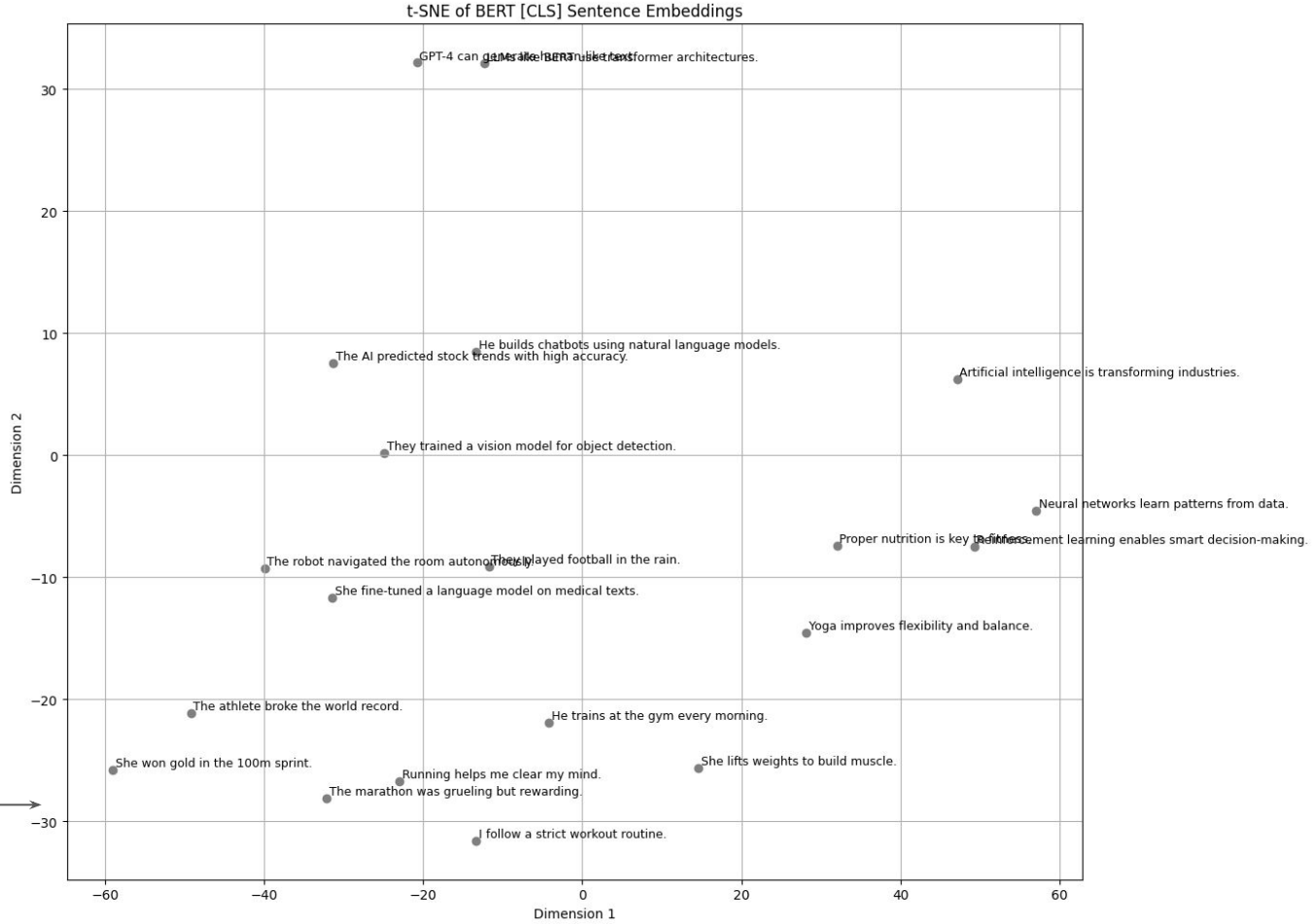
Let's extend the vectorization concept to one sentence.

Previously:
One word → 1 vector

In this slide:
1 sentence → 1 vector

```
sentences = [
    "He trains at the gym every morning.",
    "The marathon was grueling but rewarding.",
    "She lifts weights to build muscle.",
    "They played football in the rain.",
    "Running helps me clear my mind.",
    "The athlete broke the world record.",
    "I follow a strict workout routine.",
    "Yoga improves flexibility and balance.",
    "She won gold in the 100m sprint.",
    "Proper nutrition is key to fitness.",
    "Artificial intelligence is transforming industries.",
    "He builds chatbots using natural language models.",
    "Neural networks learn patterns from data.",
    "GPT-4 can generate human-like text.",
    "The robot navigated the room autonomously.",
    "They trained a vision model for object detection.",
    "Reinforcement learning enables smart decision-making.",
    "She fine-tuned a language model on medical texts.",
    "The AI predicted stock trends with high accuracy.",
    "LLMs like BERT use transformer architectures."
]
```

Plot the embeddings in 2d space via tsne

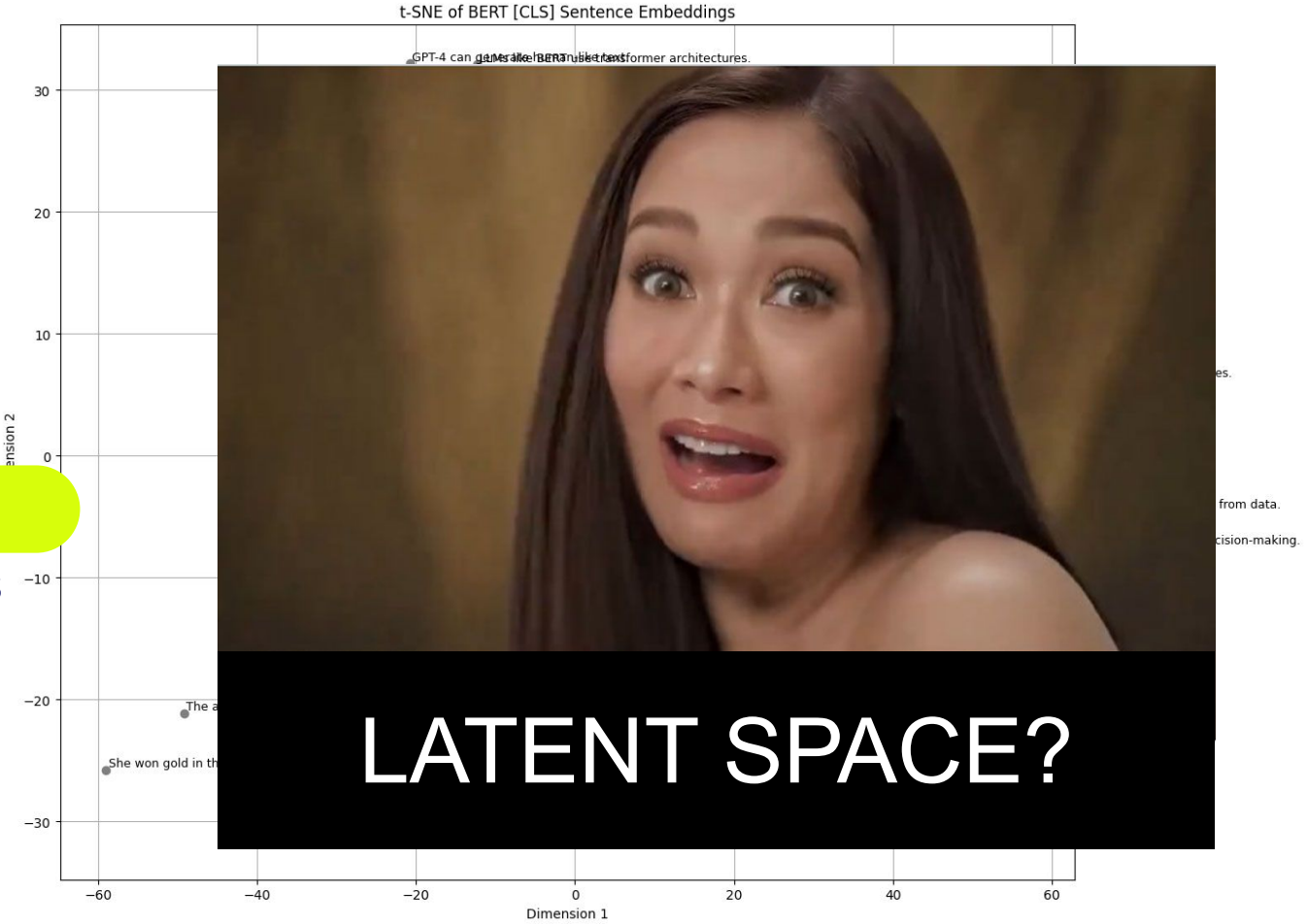


Embeddings live in the model's **latent space**

[Did you know]

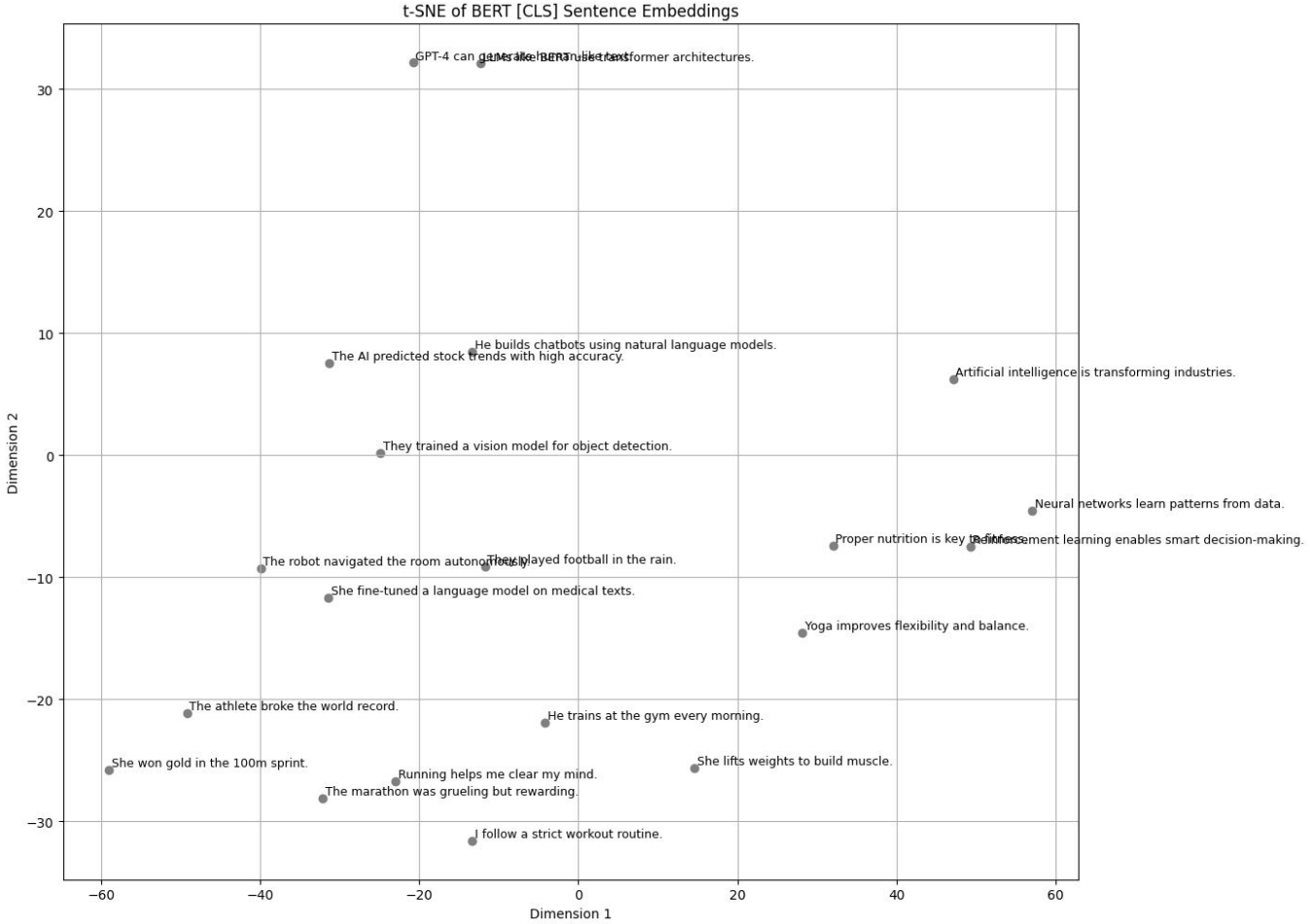
Embeddings live in the model's latent space

— where meaning is encoded in vector form.



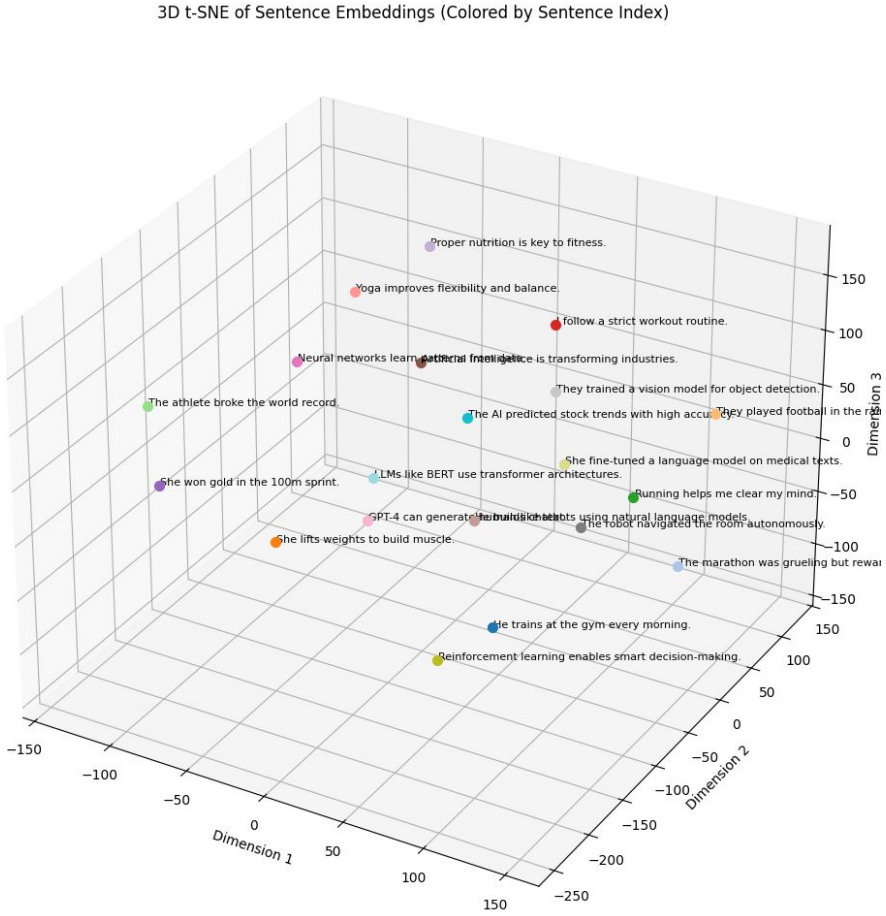
Embeddings live in the model's **latent space** (in 2d)

```
sentences = [
    "He trains at the gym every morning.",
    "The marathon was grueling but rewarding.",
    "She lifts weights to build muscle.",
    "They played football in the rain.",
    "Running helps me clear my mind.",
    "The athlete broke the world record.",
    "I follow a strict workout routine.",
    "Yoga improves flexibility and balance.",
    "She won gold in the 100m sprint.",
    "Proper nutrition is key to fitness.",
    "Artificial intelligence is transforming industries.",
    "He builds chatbots using natural language models.",
    "Neural networks learn patterns from data.",
    "GPT-4 can generate human-like text.",
    "The robot navigated the room autonomously.",
    "They trained a vision model for object detection.",
    "Reinforcement learning enables smart decision-making.",
    "She fine-tuned a language model on medical texts.",
    "The AI predicted stock trends with high accuracy.",
    "LLMs like BERT use transformer architectures."
]
```



Embeddings live in the model's **latent space** (in 3d)

```
sentences = [
    "He trains at the gym every morning.",
    "The marathon was grueling but rewarding.",
    "She lifts weights to build muscle.",
    "They played football in the rain.",
    "Running helps me clear my mind.",
    "The athlete broke the world record.",
    "I follow a strict workout routine.",
    "Yoga improves flexibility and balance.",
    "She won gold in the 100m sprint.",
    "Proper nutrition is key to fitness.",
    "Artificial intelligence is transforming industries.",
    "He builds chatbots using natural language models.",
    "Neural networks learn patterns from data.",
    "GPT-4 can generate human-like text.",
    "The robot navigated the room autonomously.",
    "They trained a vision model for object detection.",
    "Reinforcement learning enables smart decision-making.",
    "She fine-tuned a language model on medical texts.",
    "The AI predicted stock trends with high accuracy.",
    "LLMs like BERT use transformer architectures."
]
```



d

Embeddings live in the model's **latent space (in embedding Dimension)**

```
sentences = [
    "He trains at the gym every morning.",
    "The marathon was grueling but rewarding.",
    "She lifts weights to build muscle.",
    "They played football in the rain.",
    "Running helps me clear my mind.",
    "The athlete broke the world record.",
    "I follow a strict workout routine.",
    "Yoga improves flexibility and balance.",
    "She won gold in the 100m sprint.",
    "Proper nutrition is key to fitness.",
    "Artificial intelligence is transforming industries.",
    "He builds chatbots using natural language models.",
    "Neural networks learn patterns from data.",
    "GPT-4 can generate human-like text.",
    "The robot navigated the room autonomously.",
    "They trained a vision model for object detection.",
    "Reinforcement learning enables smart decision-making.",
    "She fine-tuned a language model on medical texts.",
    "The AI predicted stock trends with high accuracy.",
    "LLMs like BERT use transformer architectures."
]
```

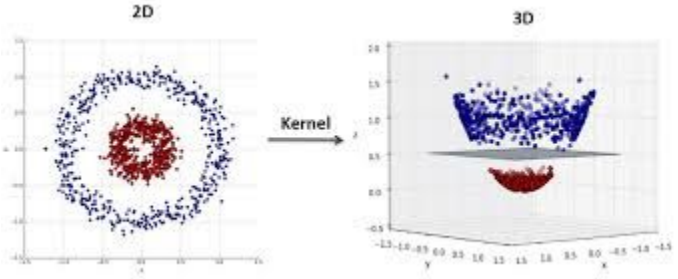
Model	Hidden Size (Embedding Dim)
BERT-base	768
BERT-large	1024
DistilBERT	768
RoBERTa-base	768
RoBERTa-large	1024
GPT-2 (small)	768
GPT-2 (medium)	1024
GPT-2 (large)	1280
GPT-2 (XL)	1600
GPT-3 (Davinci)	12288

d

Why is 2d or 3d not enough?

```
sentences = [
    "He trains at the gym every morning.",
    "The marathon was grueling but rewarding.",
    "She lifts weights to build muscle.",
    "They played football in the rain.",
    "Running helps me clear my mind.",
    "The athlete broke the world record.",
    "I follow a strict workout routine.",
    "Yoga improves flexibility and balance.",
    "She won gold in the 100m sprint.",
    "Proper nutrition is key to fitness.",
    "Artificial intelligence is transforming industries.",
    "He builds chatbots using natural language models.",
    "Neural networks learn patterns from data.",
    "GPT-4 can generate human-like text.",
    "The robot navigated the room autonomously.",
    "They trained a vision model for object detection.",
    "Reinforcement learning enables smart decision-making.",
    "She fine-tuned a language model on medical texts.",
    "The AI predicted stock trends with high accuracy.",
    "LLMs like BERT use transformer architectures."
]
```

Model	Hidden Size (Embedding Dim)
BERT-base	768
BERT-large	1024
DistilBERT	768
RoBERTa-base	768
RoBERTa-large	1024
GPT-2 (small)	768
GPT-2 (medium)	1024
GPT-2 (large)	1280
GPT-2 (XL)	1600
GPT-3 (Davinci)	12288



d

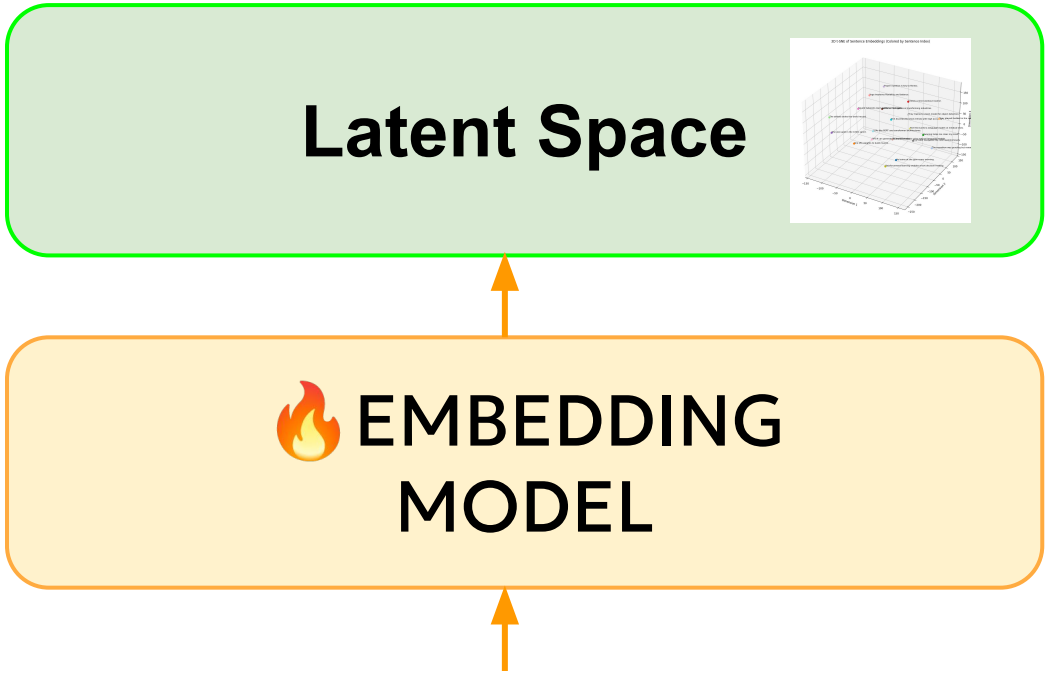
Borrowing the concept of the kernel trick, we can explain why embeddings work: **by moving text into a higher-dimensional space, it's easier to separate meanings** just like how the kernel trick separates non-linear patterns by projecting data into a space where it's linearly separable."

Embeddings live in the model's **latent space**

[Did you know]

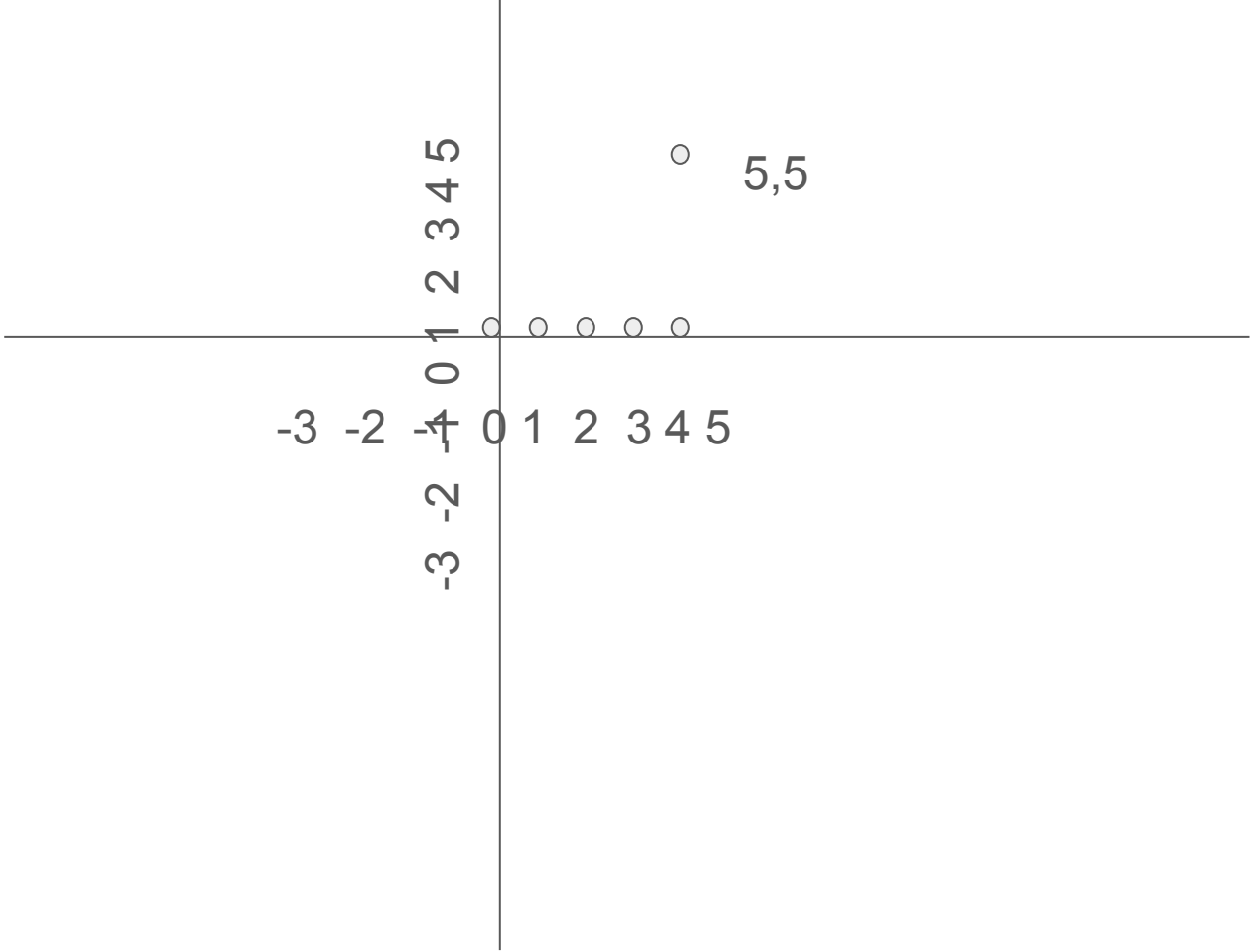
Embeddings live in the model's latent space

— where meaning is encoded in vector form.



Activity

https://colab.research.google.com/drive/1f8Bcqo_D0fIkrSzlYXVCvN1UISZphBJ4#scrollTo=ouzWVBQjz3tT

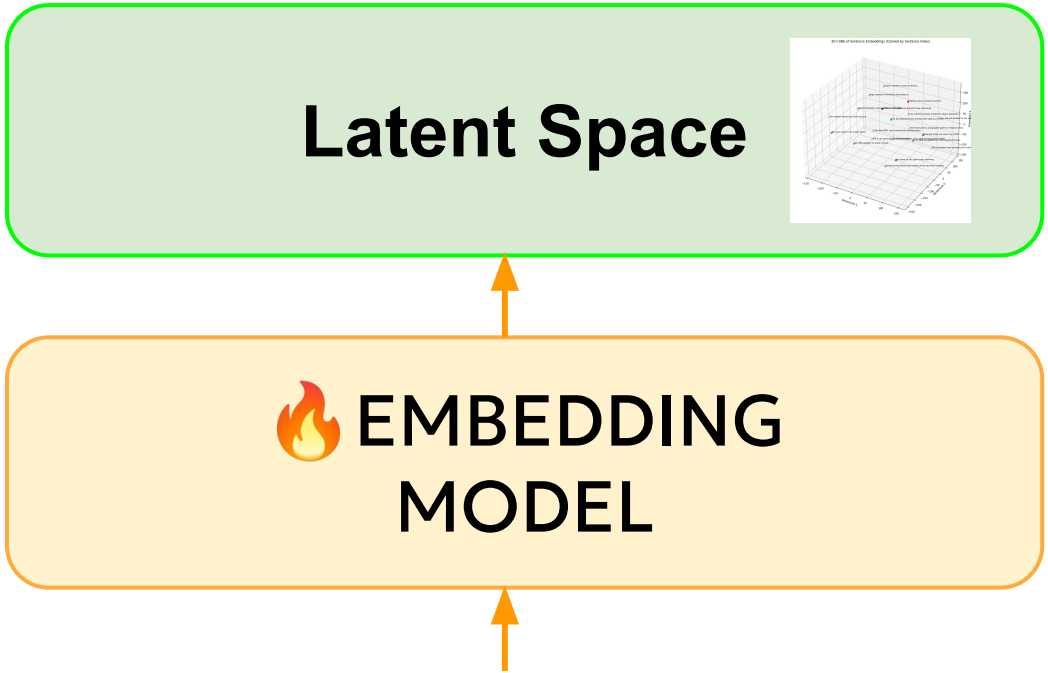

5,5
-3 -2 -1 0 1 2 3 4 5
5 4 3 2 1 0 -1 -2 -3

Embeddings live in the model's **latent space**

[Did you know]

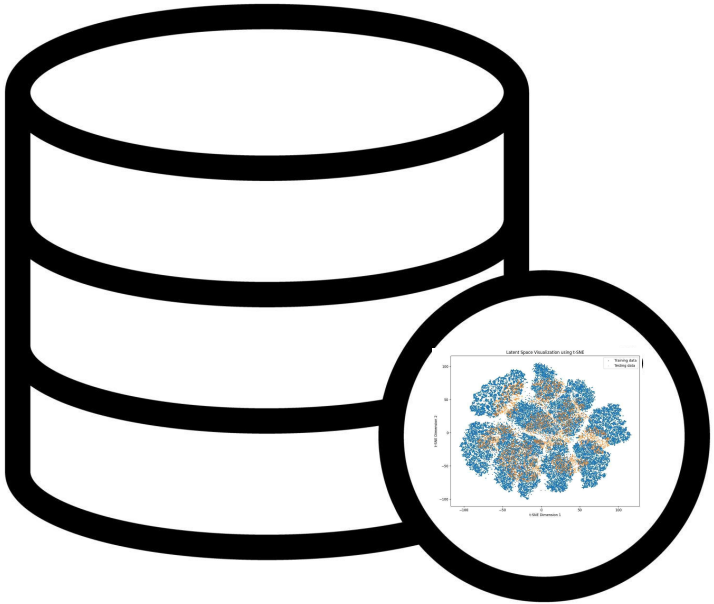
Embeddings live in the model's latent space

— where meaning is encoded in vector form.

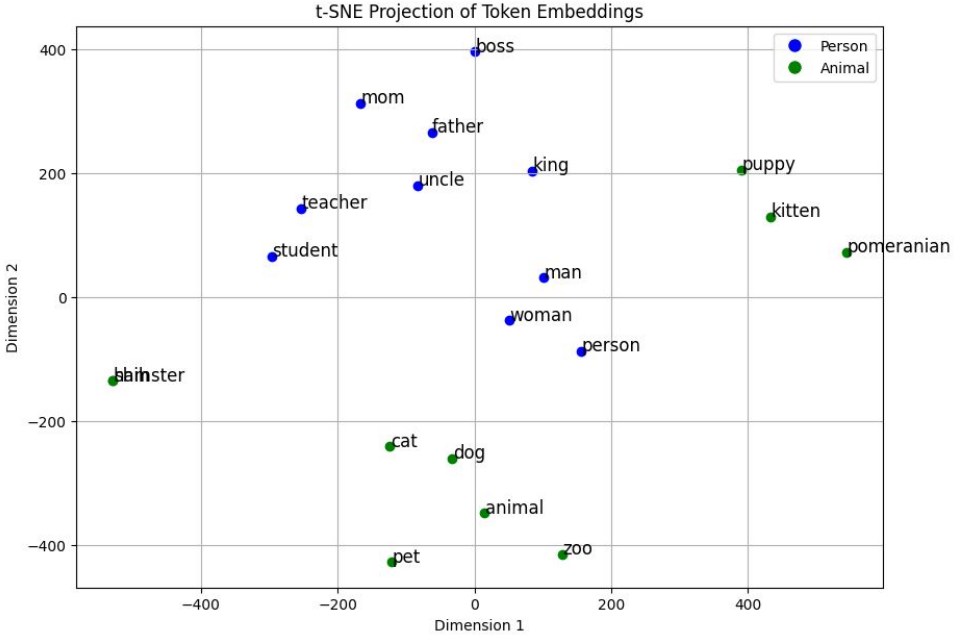


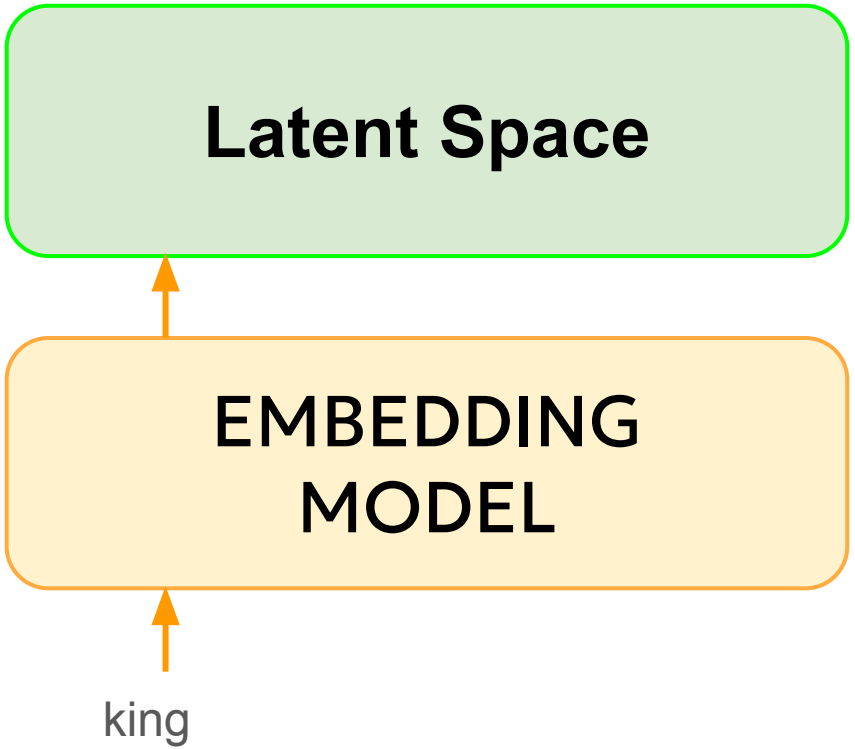
Nikko Carlo Yabut

Saving the **Latent Space** in a **Vector Database**

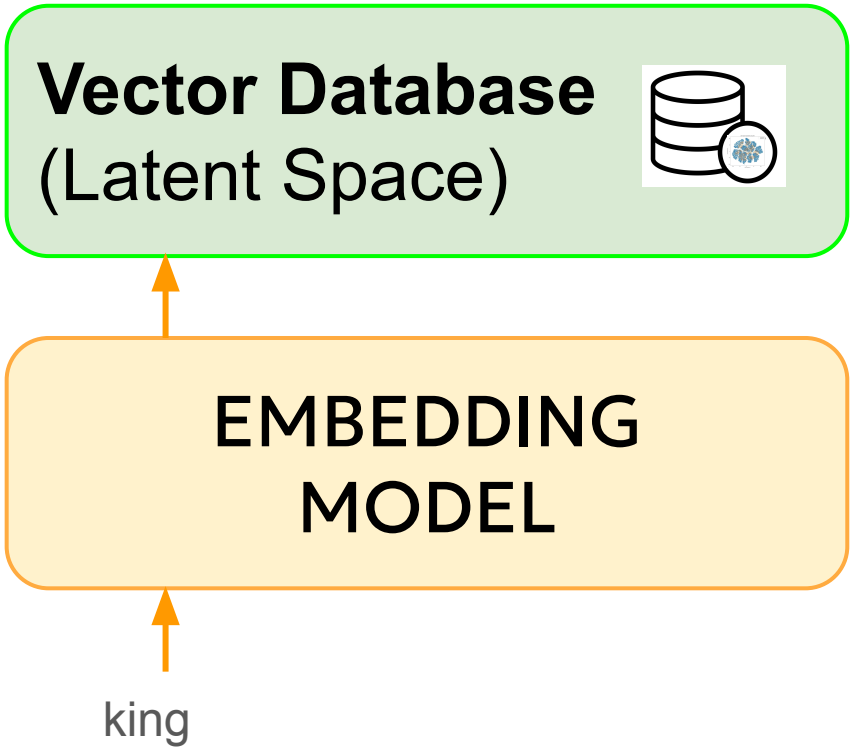


Saving the Latent Space in a Vector Database



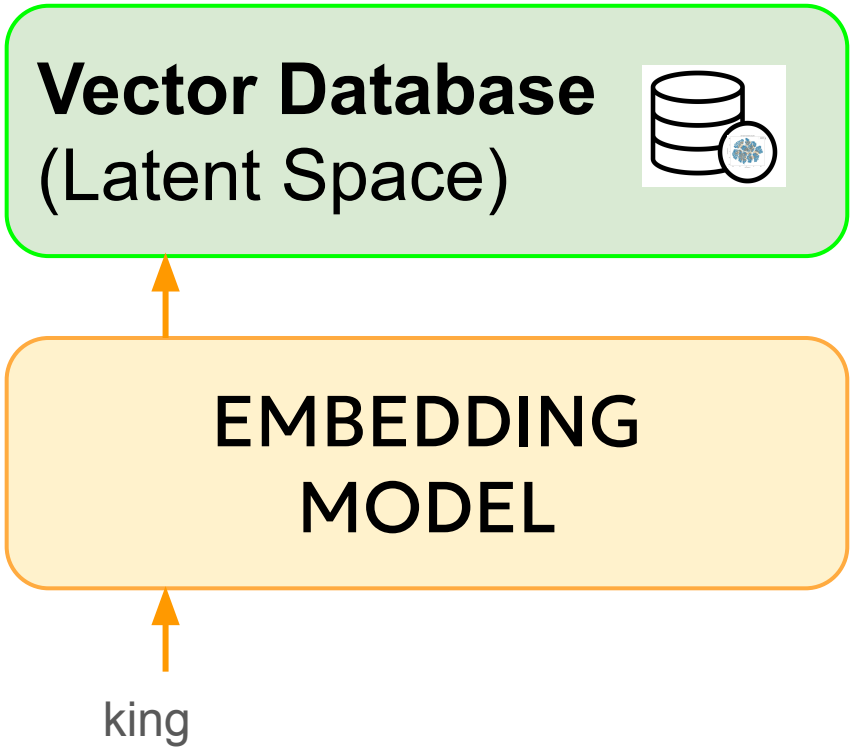


Saving the Latent Space in a Vector Database

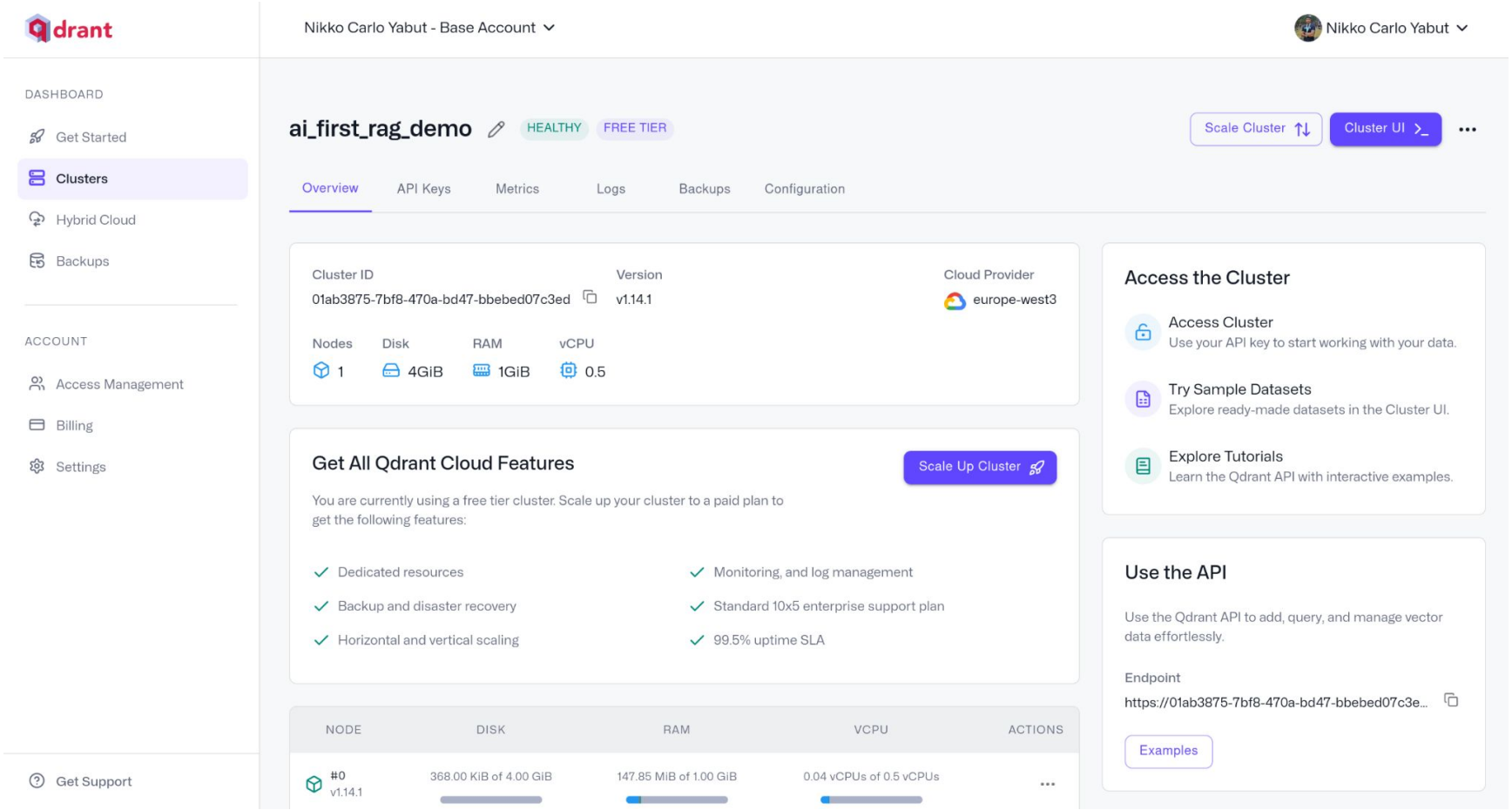


Saving the Latent Space in a Vector Database

Name	Open Source	Scalable	Cloud-hosted	Notes
FAISS	✅	❌	❌	Local-only, blazing fast
Qdrant	✅	✅	✅	Rust-based, modern APIs
Pinecone	❌	✅	✅	API-only, closed-source
Weaviate	✅	✅	✅	ML pipelines, hybrid search
Chroma	✅	❌	❌	Lightweight, local-only
Vespa	✅	✅	✅	Enterprise-grade, complex

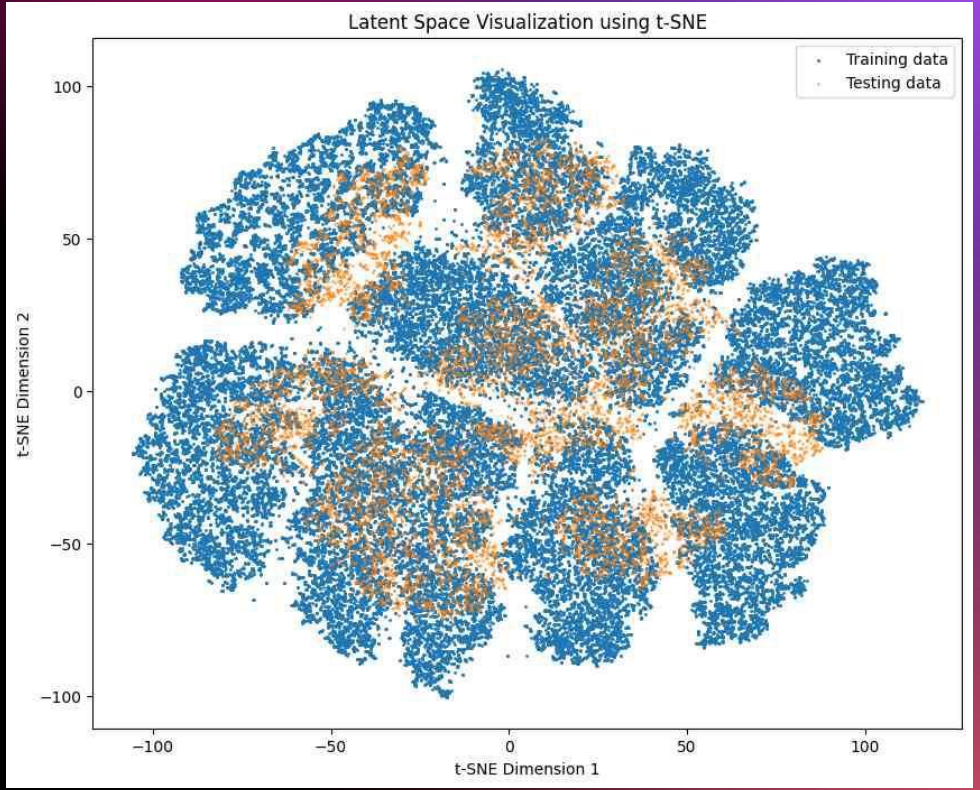


Saving the Latent Space in a Vector Database



Nikko Carlo Yabut

Retrieving Data from the **Latent Space**



Why can't we use MySQL to search for similar text?

Why can't we use **MySQL** to search for similar text?

Because MySQL (and traditional databases) are built for **exact matches** or **basic filters** — not understanding.

---

f you ask:
`SELECT * FROM documents WHERE text = 'climate change'`
You'll get only documents that contain exactly that phrase — nothing semantically related like "global warming" or "environmental shifts."

They operate on surface-level text, **not meaning.**

Traditional DBs use string comparison or keyword search (e.g., LIKE, REGEXP)

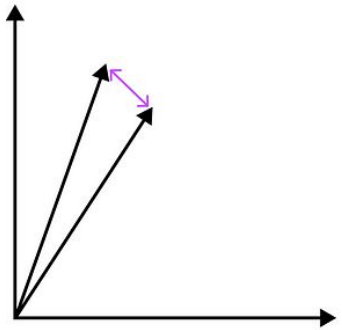
Text A: "The cat sat on the mat."
Text B: "A feline rested on the rug."

A MySQL `LIKE '%cat%'` query would miss Text B entirely

"If we can't use LIKE or REGEX, how can we perform a search?"

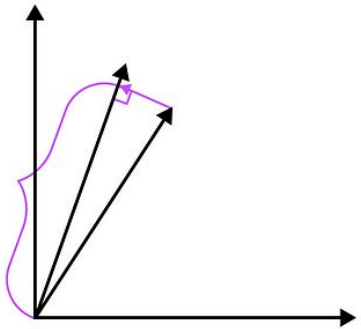
Euclidean Distance

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$



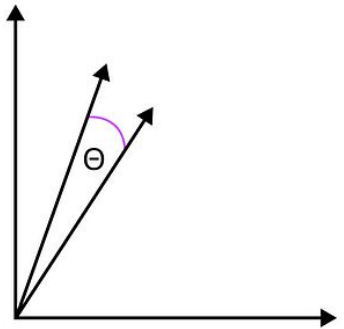
Inner Product

$$a \cdot b = \sum_{i=1}^{n} a_i b_i$$

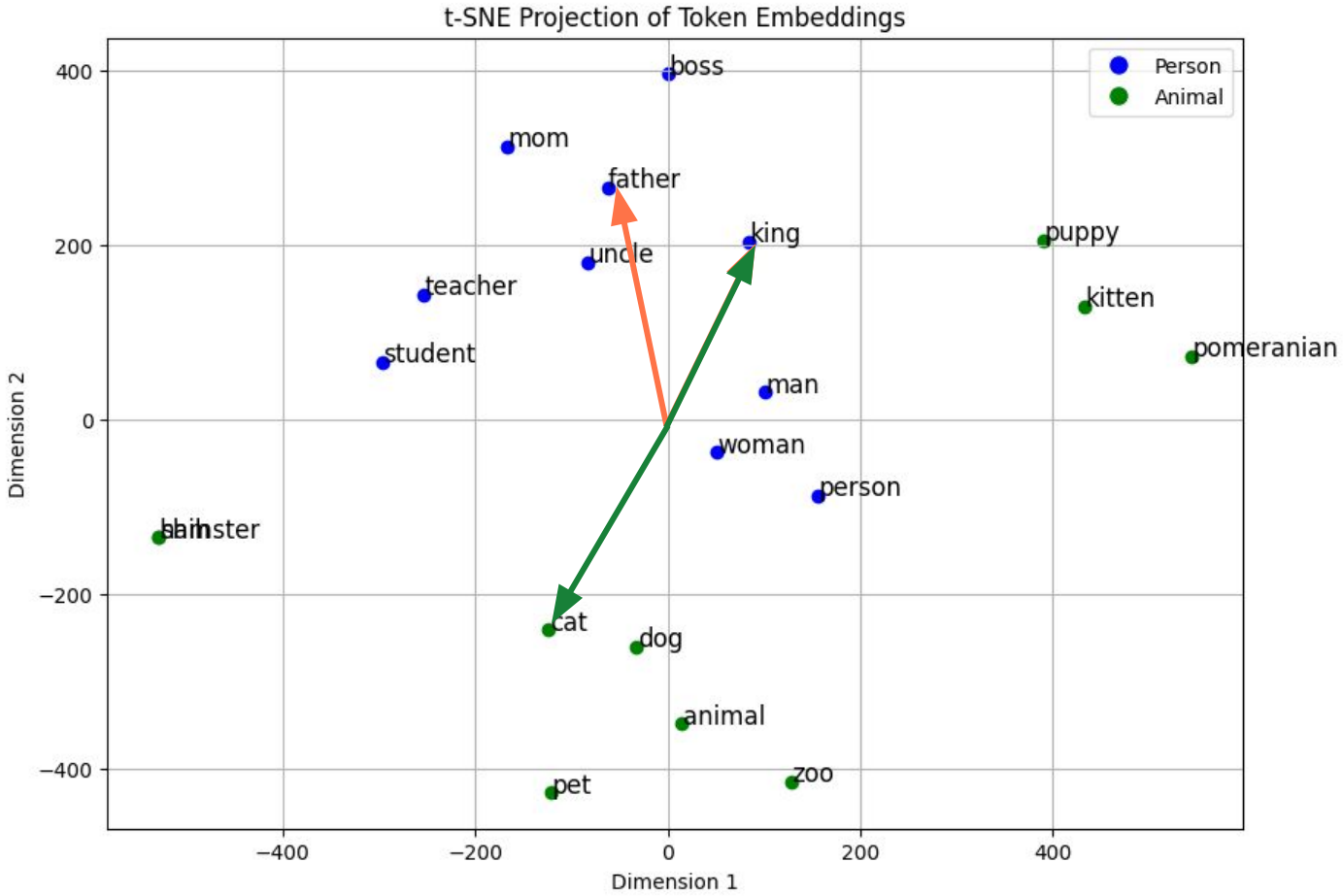


Cosine Similarity

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

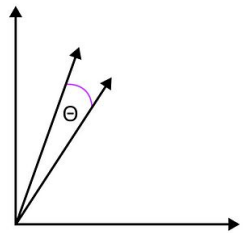


The Cosine Similarity

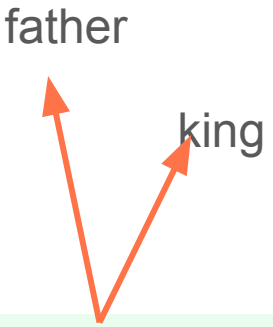


Cosine Similarity

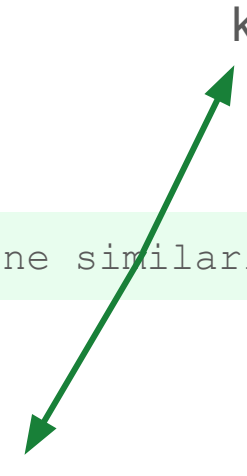
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$



The Cosine Similarity



Cosine similarity = 0.37



Cosine similarity = 0.25

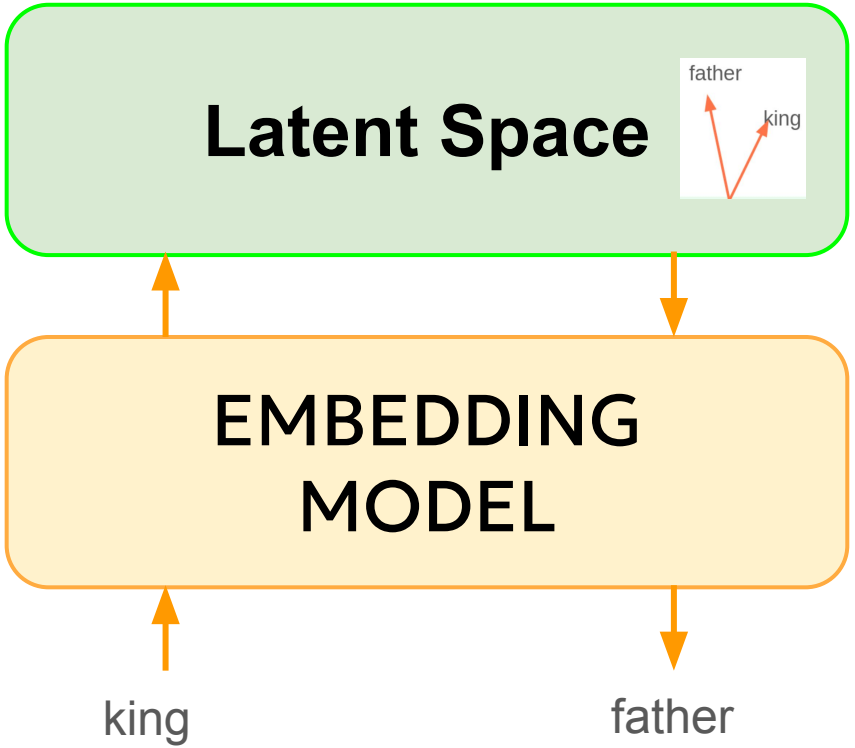
Cosine Similarity Matrix:

	king	father	teache	stude
king	1.00	0.37	0.26	0.2
father	0.37	1.00	0.40	0.3
teacher	0.26	0.40	1.00	0.5
student	0.25	0.30	0.51	1.0
man	0.35	0.35	0.22	0.2
woman	0.28	0.34	0.33	0.3
person	0.27	0.28	0.30	0.3
mom	0.22	0.49	0.40	0.2
boss	0.29	0.35	0.32	0.2
uncle	0.31	0.56	0.39	0.2
dog	0.23	0.30	0.31	0.3
cat	0.25	0.27	0.29	0.3
puppy	0.23	0.30	0.39	0.3
kitten	0.23	0.27	0.39	0.3
pomeranian	0.24	0.29	0.41	0.3
shih	0.19	0.21	0.28	0.2
zoo	0.21	0.19	0.30	0.2
animal	0.21	0.26	0.33	0.3
pet	0.29	0.29	0.27	0.3
hamster	0.19	0.21	0.28	0.2

Retrieving the Data from the Latent Space

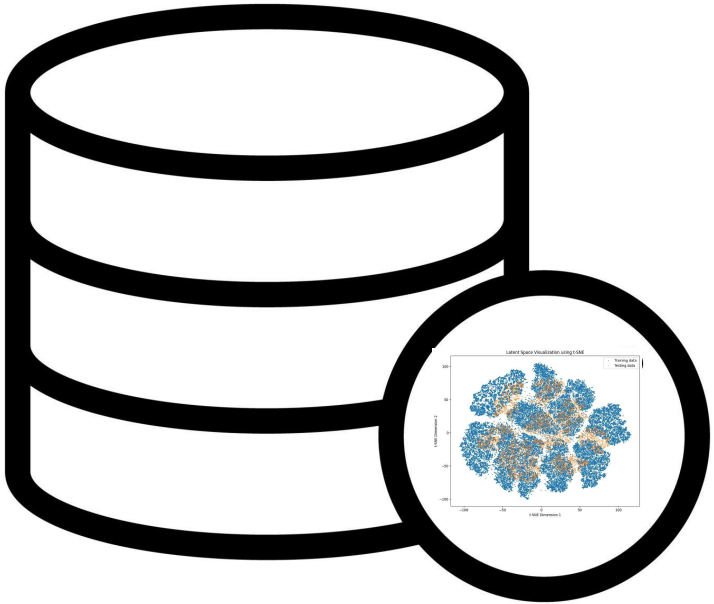
Through embedding models, we can **convert both our documents and queries into vectors** that live in a shared latent space.

By using similarity measures like cosine similarity, we can **identify and then retrieve the most semantically relevant matches**—even if the wording is completely different.



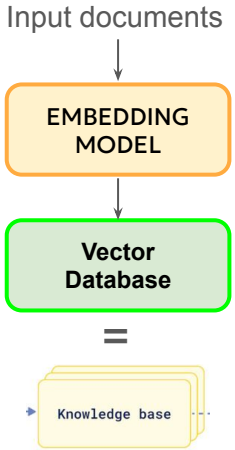
Nikko Carlo Yabut

RAG



RAG (Creating the Knowledge Base)

Note:
If the document is large, it needs to be **preprocessed into smaller chunks** so each piece can be meaningfully embedded and retrieved during search or generation.



🔹 **Big chunks**
Pros: More context
Cons: Harder to retrieve relevant info, may dilute meaning

🔹 **Small chunks**
Pros: Precise retrieval
Cons: May lack context, fragmented meaning

🧾 **Small Chunk (Good for Transactions):**

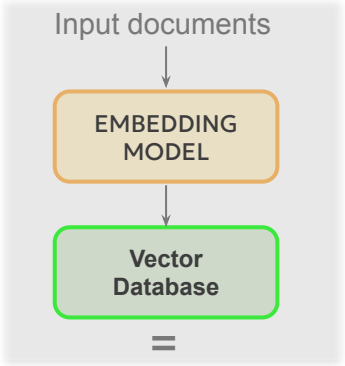
```
"Order #12345 placed on June 5 for $49.99."
```

📖 **Big Chunk (Good for Stories):**

```
"Maria walked into the store feeling hopeful. After browsing through
several aisles, she picked a book that reminded her of home. As she
reached the counter, a familiar song played, bringing back childhood
memories."
```

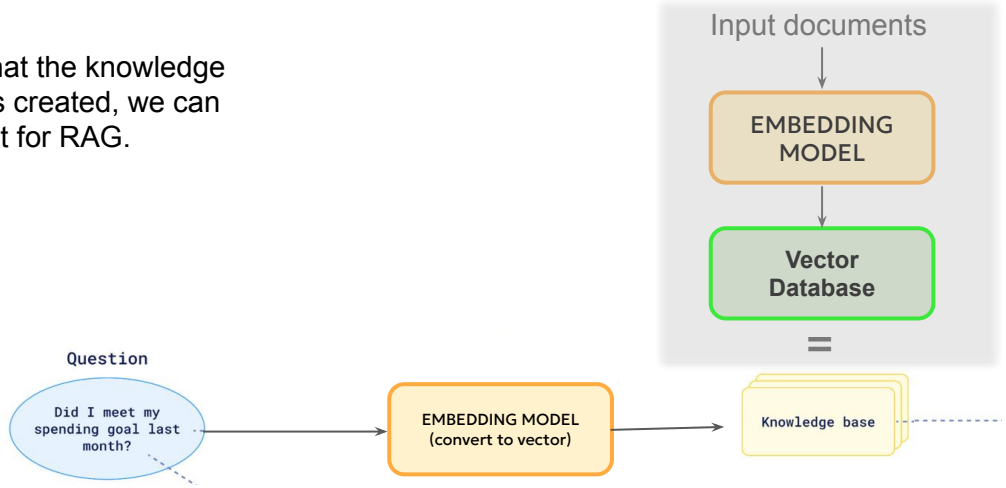
RAG (Inference)

Now that the knowledge base is created, we can query it for RAG.



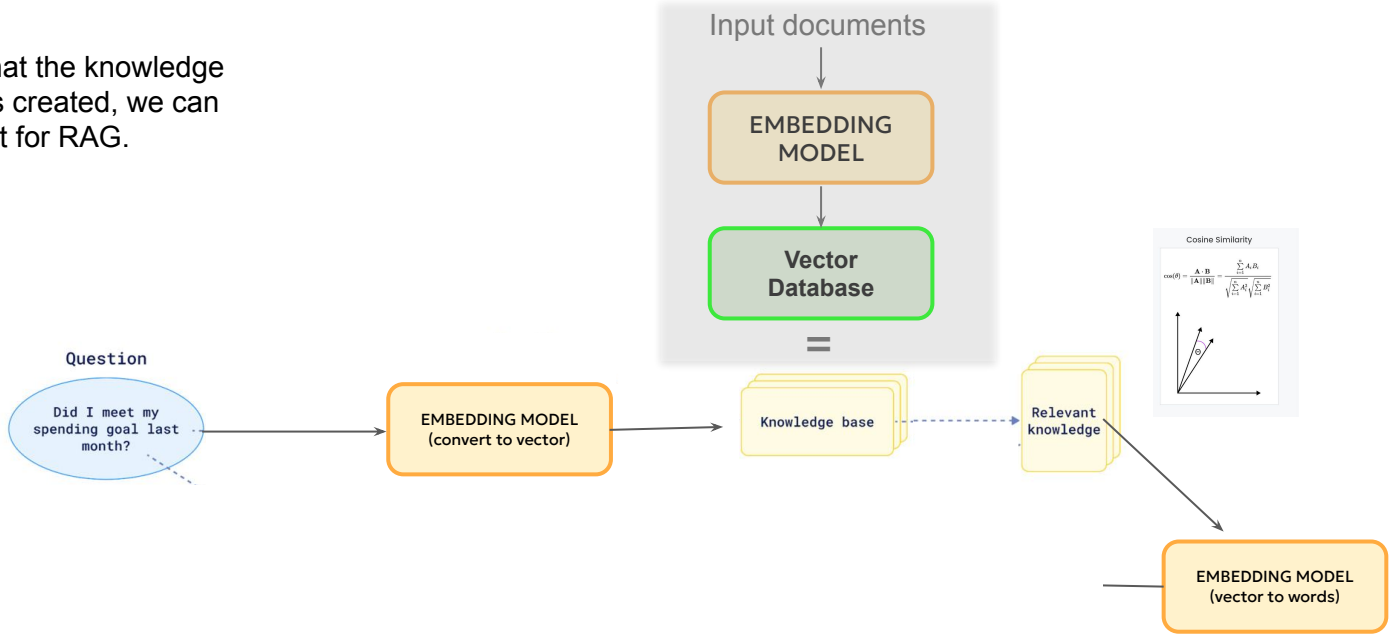
RAG (Inference)

Now that the knowledge base is created, we can query it for RAG.



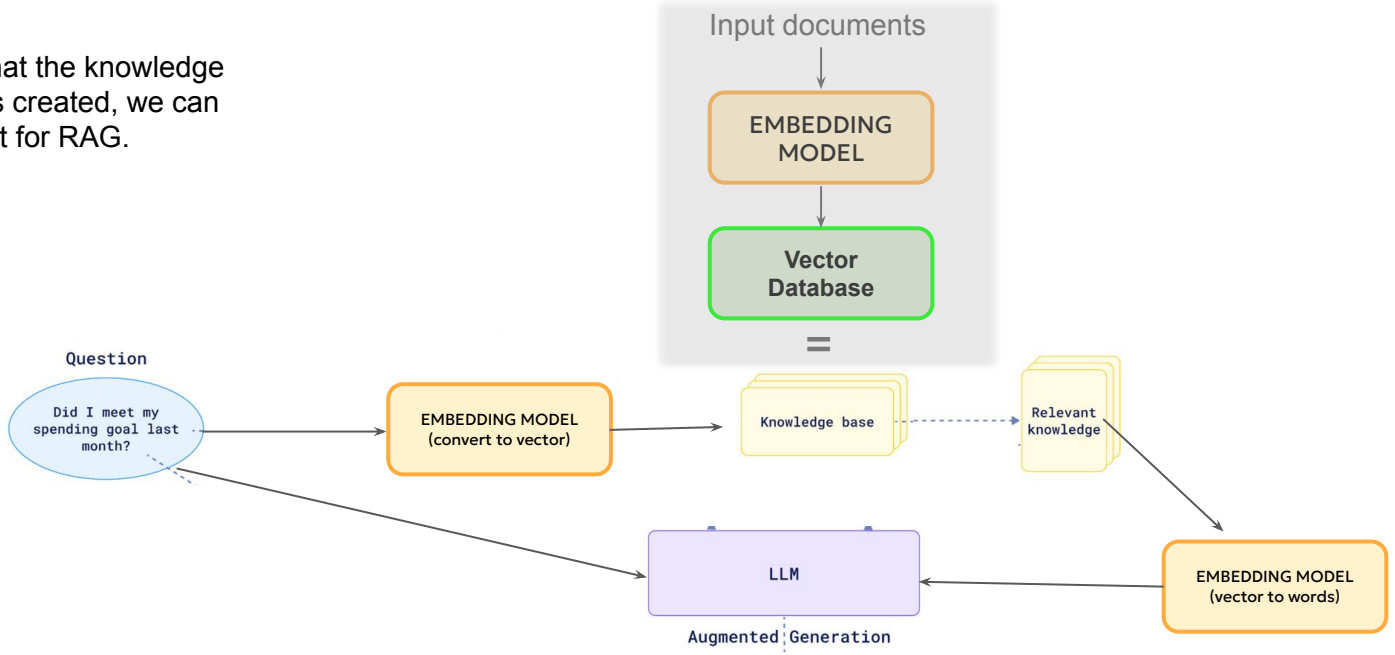
RAG (Inference)

Now that the knowledge base is created, we can query it for RAG.



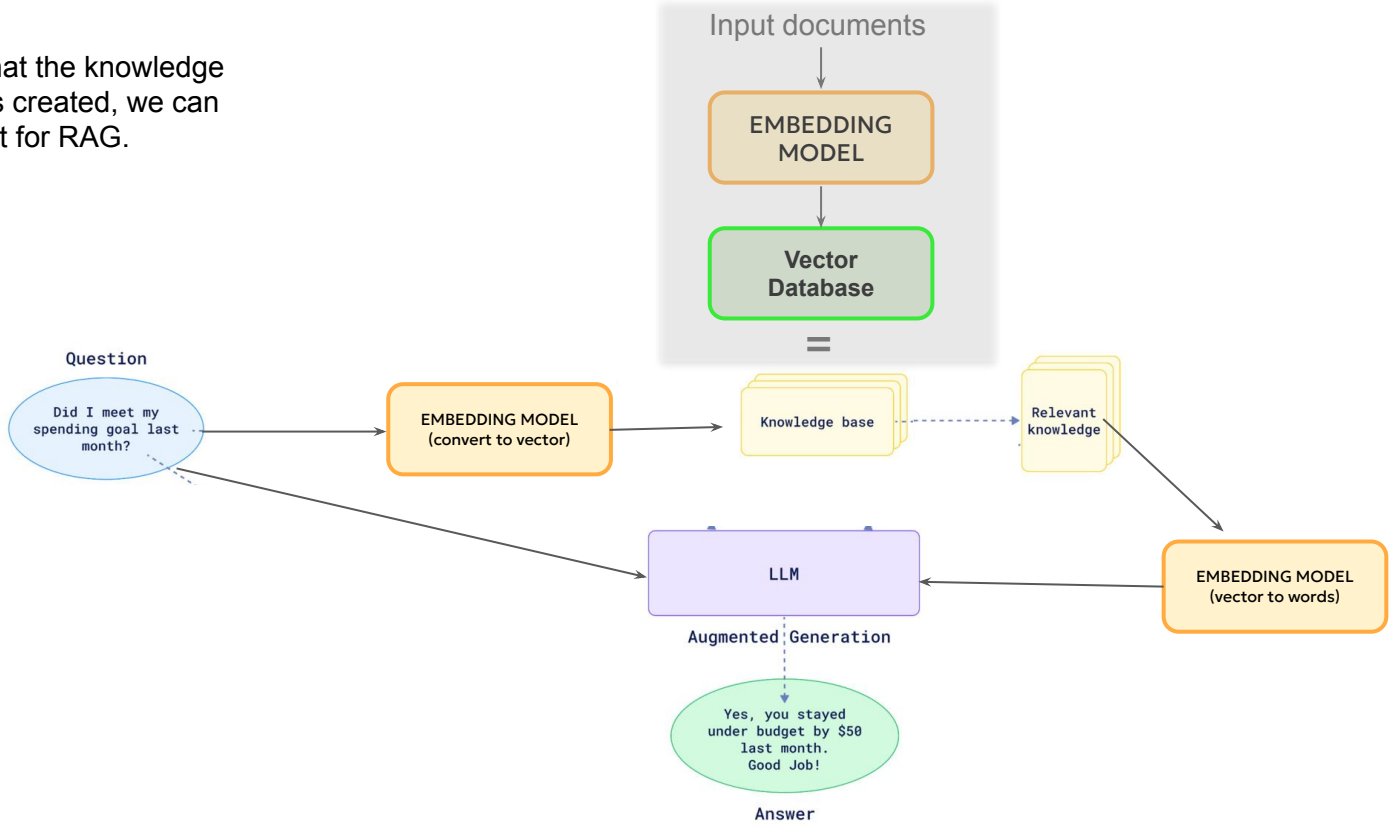
RAG (Inference)

Now that the knowledge base is created, we can query it for RAG.



RAG (Inference)

Now that the knowledge base is created, we can query it for RAG.



What's on your mind today?

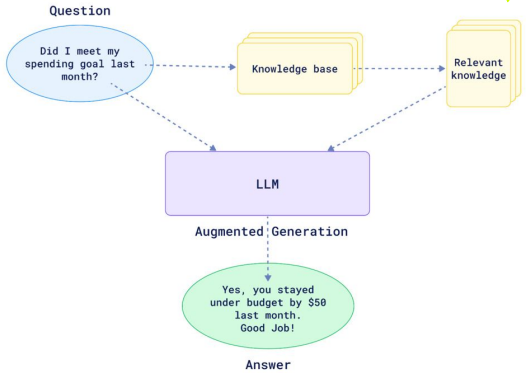
If you were to build a chatbot that could answer questions about your company's documents, how would you do it?

+ Tools

Better Approach:

"Instead of fine-tuning the model every time the knowledge base updates, what if we just store the documents in a database — and let the model retrieve the relevant chunks during inference?"

RAG



- Use a frozen, pretrained model (e.g., GPT or Mistral)
- Store your documents in a **vector database** (like FAISS, Weaviate, Qdrant, etc.)
- At runtime, convert the query and your documents into **embeddings** (aka latent representations)
- Retrieve relevant documents using **semantic similarity**
- Feed them into the LLM (via context window)

Activity

Set Up Qdrant Cloud Account (Before Class)

1. Go to https://cloud.qdrant.io
2. Create an account (free tier is enough)
3. Create a new cluster (select free, small, single region)
4. Inside your cluster, create a **collection**
5. **Copy the following info:**
    - `API Key`
    - `Cluster URL` (something like `https://xxxxxxxxxxx.qdrant.tech`)

Cont activity in colab:
https://colab.research.google.com/drive/13LLD5bcfdKadjz0wrJNRjY2mRS97hsPO?usp=sharing

Business Case Ideation

Create a Business Case utilizing RAG to solve your chosen problem.

1. Define the problem
- Identify 1–2 clear pain points.
  ⚠️ Focus on content-heavy or info lookup problems
- Ex: *“Agents waste time finding answers,” “Students can’t digest long readings.”*

2. Who is your user?
- Who feels this problem?
  Ex: *Support reps*, *Students*, *HR staff*, etc.

3. How RAG will help? What does the tool do?
- What does the tool **do**? Ex: *“Answers HR questions from uploaded handbook”*

4. What are its main features?
   a. Upload knowledge base
   b. User types a question
   c. App returns grounded answer
5. Build your own streamlit app and do a demo

Nikko Carlo Yabut

LLM's Autoregression

Intro to Agentic AI

Models

OPEN-SOURCE

LLM
- **LLaMA 2** (by Meta)
- **OpenLLAMA** (by Berkeley AI)
- **Gemma** (by Google)
- **Mixtral** (by Mistral AI)
- **Mistral** (by Mistral AI)
- **Dolly** (by Databricks)
- **Falcon** (by TII)
- **DeepSeek** (by DeepSeek AI)

Vision
- **SAM** (by Meta AI)
- **Detectron2** (by Facebook AI R)
- **MMDetection** (by OpenMMlab)
- **YOLOv8** (by Ultralytics)

VLM
- **MiniGPT-4**
- **BLIP-2** (by Anthropic)
- **BLIP** (by Salesforce)
- **CLIP** (by Open AI)

Audio
- **ESPnet**
- **Wav2Vec**
- **Whisper**

Word2Vec
- **Word2Vec** (by PyTorch)
- **Word2Vec** (by Gensim)
- **FastTeext** (by Facebook)
- **SentenceTransformer** (by Huggingface)

CLOSED-SOURCE

LLM
- **GPT-4** (by OpenAI)
- **Claude 3** (by Anthropic)
- **Gemini** (by Google DeepMind)
- **Command R+** (by Cohere)
- **Ernie Bot** (by Baidu)
- **Qwen** (by Alibaba)
- **SenseChat** (by SenseTime)
- **Grok** (by xAI)

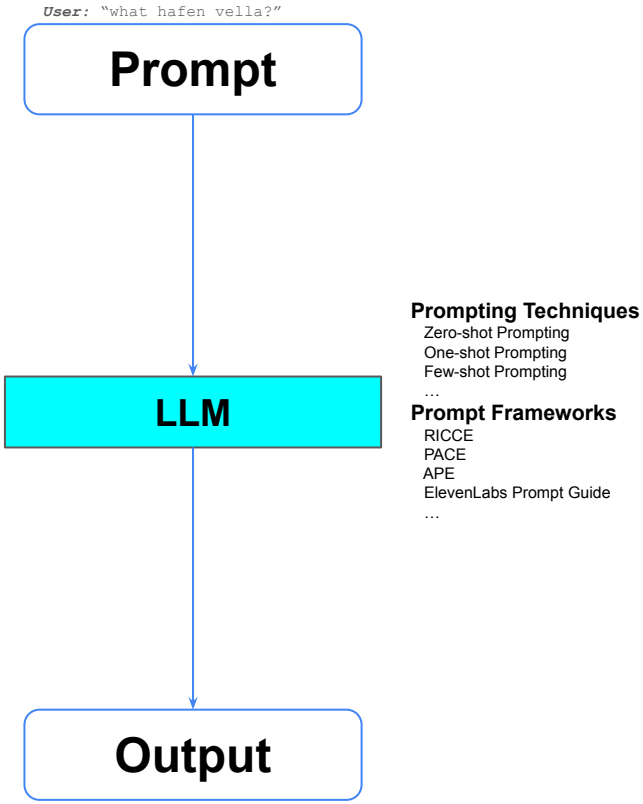
Vision
- **Amazon Rekognition** (by AWS)

VLM
- **GPT-4V** (by OpenAI)
- **Claude 4 Opus** (by Anthropic)
- **Gemini** (by Google DeepMind)
- **Grok Vision** (by xAI)
- **Imagen**
- **Midjourney**
- **Flux**

Audio
- **ElevenLabs**
- **Google Speech-to-Text**
- **Amazon Transcribe**
- **Microsoft Azure Speech**
- **Descript Overdub**

Word2Vec
- **OpenAI Embeddings**
- **Google Cloud Embeddings**
- **Cohere Embeddings**
- **Amazon Comprehend**

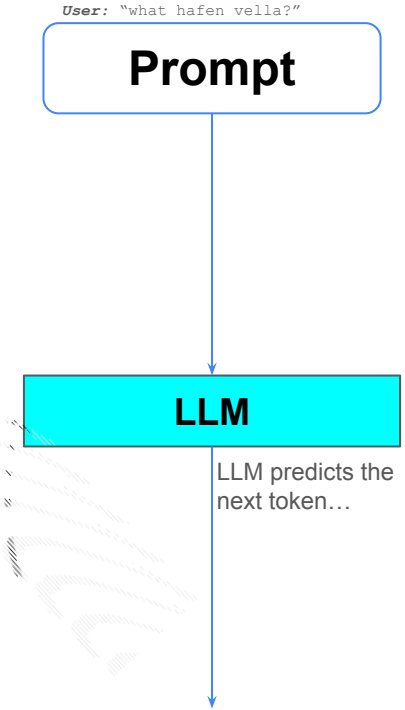
Using LLM to solve a task (**Chat Completion**)



Prompting Techniques
- Zero-shot Prompting
- One-shot Prompting
- Few-shot Prompting
- …

Prompt Frameworks
- RICCE
- PACE
- APE
- ElevenLabs Prompt Guide
- …

Using LLM to solve a task (**Chat Completion**) -Auto-regression

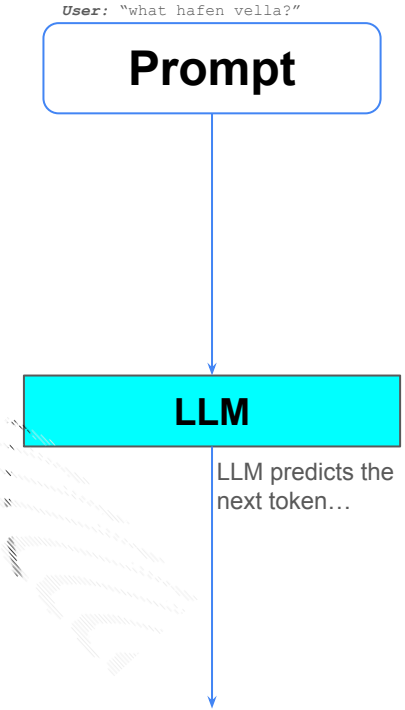


Prompting Techniques
- Zero-shot Prompting
- One-shot Prompting
- Few-shot Prompting
- …

Prompt Frameworks
- RICCE
- PACE
- APE
- ElevenLabs Prompt Guide
- …

User: "what hafen vella?"
Assistant: "The

Using LLM to solve a task (**Chat Completion**) -Auto-regression

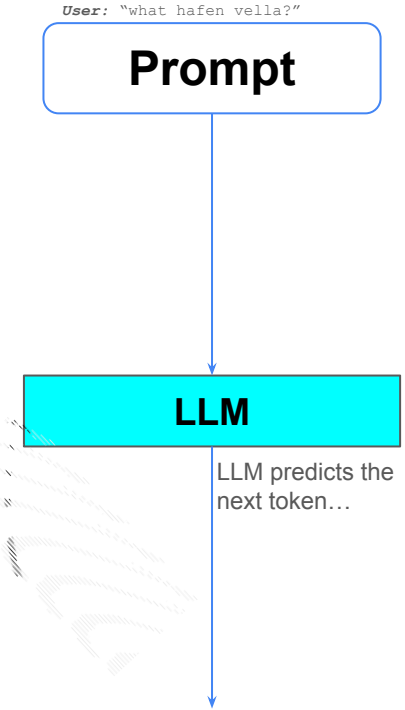


Prompting Techniques
- Zero-shot Prompting
- One-shot Prompting
- Few-shot Prompting
- …

Prompt Frameworks
- RICCE
- PACE
- APE
- ElevenLabs Prompt Guide
- …

User: "what hafen vella?"
Assistant: "The phrase

Using LLM to solve a task (**Chat Completion**) -Auto-regression

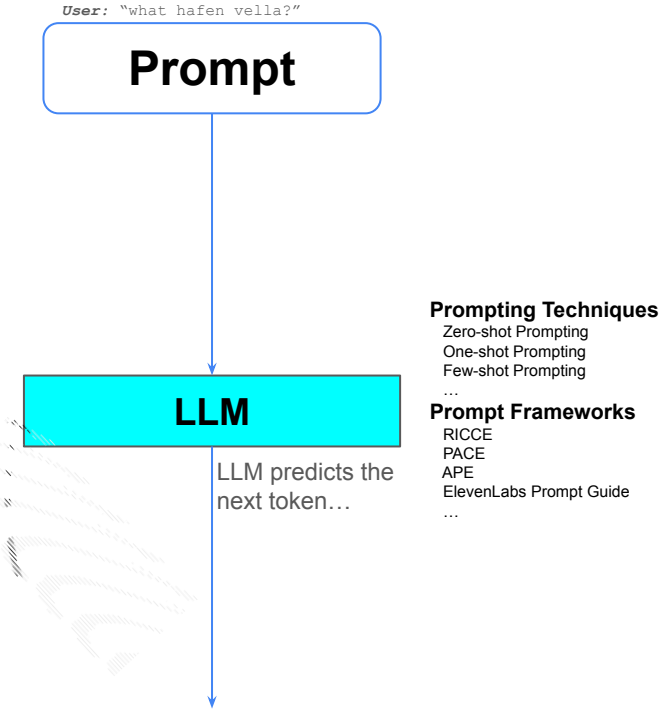


Prompting Techniques
- Zero-shot Prompting
- One-shot Prompting
- Few-shot Prompting
- …

Prompt Frameworks
- RICCE
- PACE
- APE
- ElevenLabs Prompt Guide
- …

User: "what hafen vella?"
Assistant: "The phrase "What

Using LLM to solve a task (**Chat Completion**) -Auto-regression

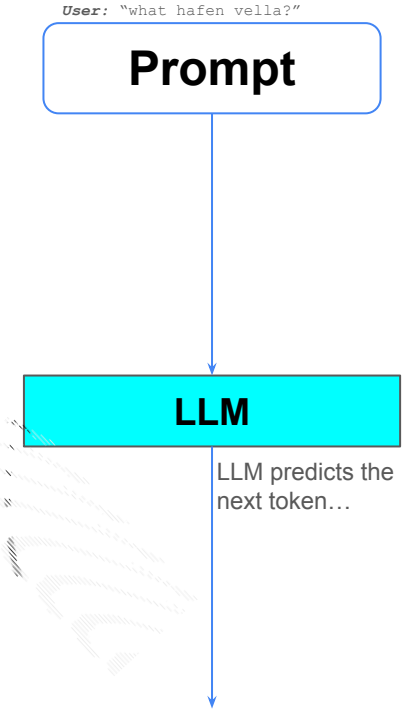


Prompting Techniques
- Zero-shot Prompting
- One-shot Prompting
- Few-shot Prompting
- …

Prompt Frameworks
- RICCE
- PACE
- APE
- ElevenLabs Prompt Guide
- …

User: "what hafen vella?"
Assistant: "The phrase "What hafen

Using LLM to solve a task (**Chat Completion**) -Auto-regression

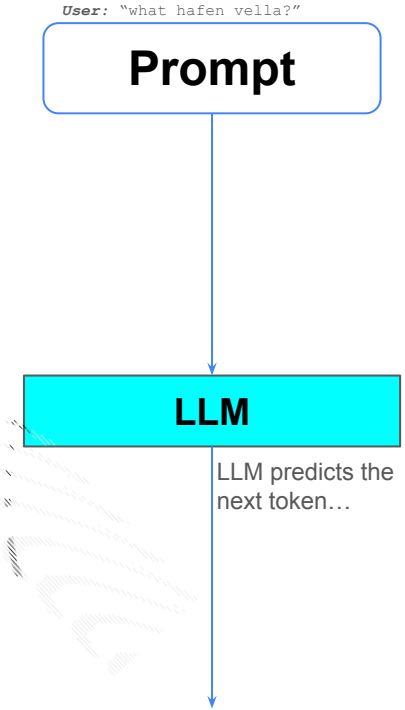


Prompting Techniques
- Zero-shot Prompting
- One-shot Prompting
- Few-shot Prompting
- …

Prompt Frameworks
- RICCE
- PACE
- APE
- ElevenLabs Prompt Guide
- …

User: "what hafen vella?"
Assistant: "The phrase "What hafen vella?"

Using LLM to solve a task (**Chat Completion**) -Auto-regression

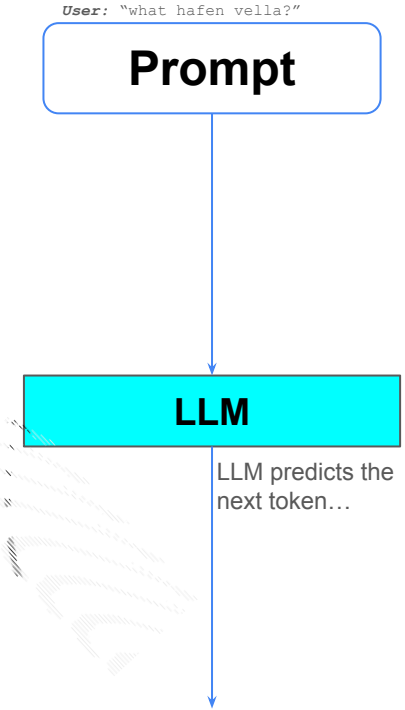


Prompting Techniques
- Zero-shot Prompting
- One-shot Prompting
- Few-shot Prompting
- …

Prompt Frameworks
- RICCE
- PACE
- APE
- ElevenLabs Prompt Guide
- …

User: "what hafen vella?"
Assistant: "The phrase "What hafen vella?" is

Using LLM to solve a task (**Chat Completion**) -Auto-regression

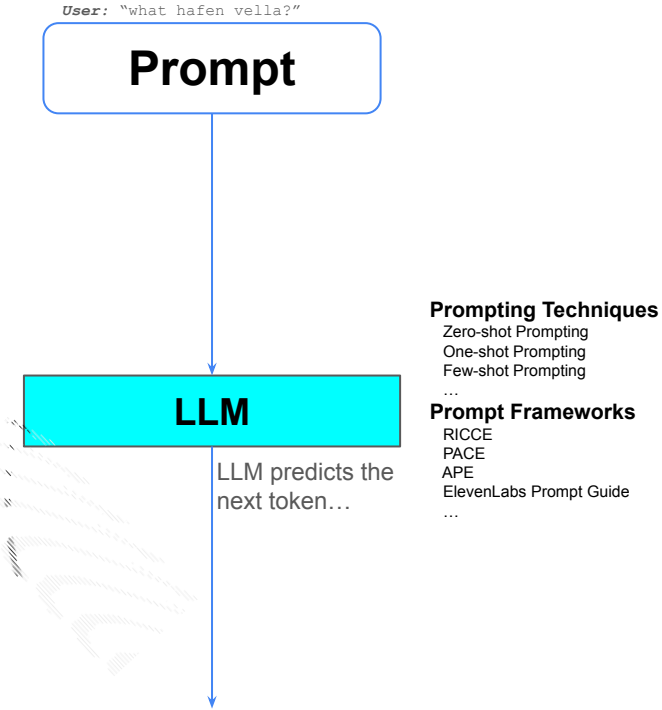


Prompting Techniques
- Zero-shot Prompting
- One-shot Prompting
- Few-shot Prompting
- …

Prompt Frameworks
- RICCE
- PACE
- APE
- ElevenLabs Prompt Guide
- …

User: "what hafen vella?"
Assistant: "The phrase "What hafen vella?" is a

Using LLM to solve a task (**Chat Completion**) -Auto-regression

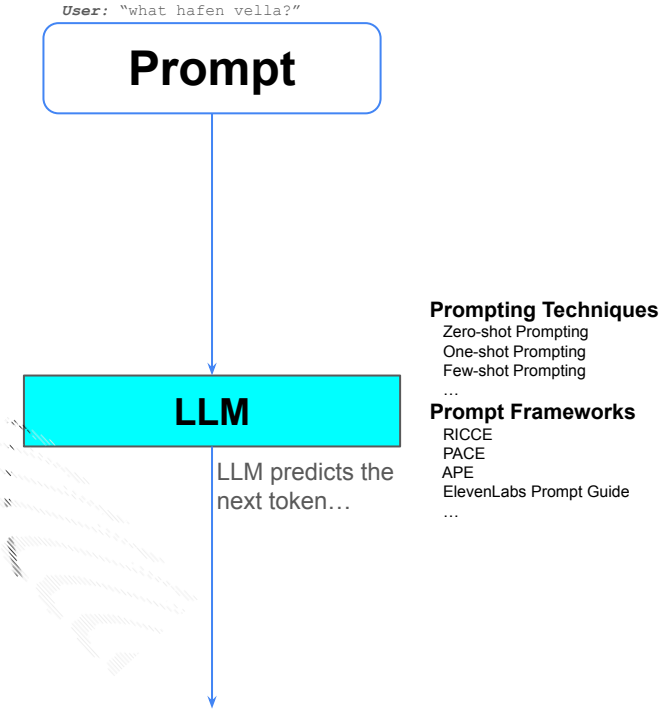


Prompting Techniques
- Zero-shot Prompting
- One-shot Prompting
- Few-shot Prompting
- …

Prompt Frameworks
- RICCE
- PACE
- APE
- ElevenLabs Prompt Guide
- …

User: "what hafen vella?"
Assistant: "The phrase "What hafen vella?" is a viral

Using LLM to solve a task (**Chat Completion**) -Auto-regression



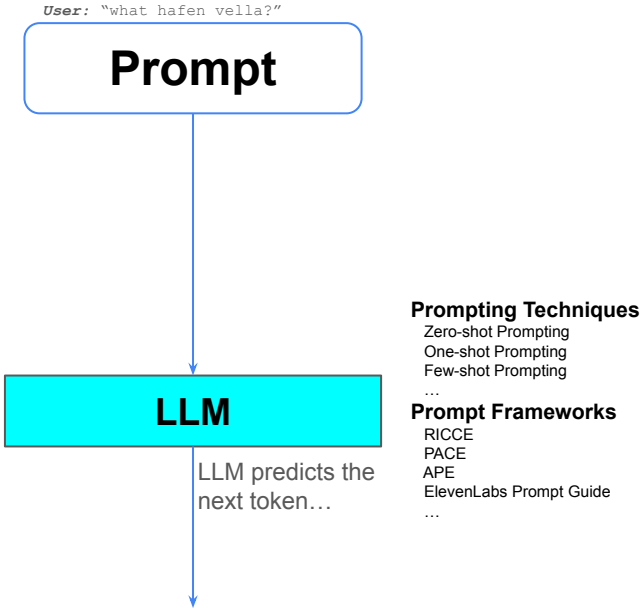
User: "what hafen vella?"
Assistant: "The phrase "What hafen vella?" is a viral meme

This **autoregressive** behavior continues token by token.

Each new token is predicted based on **all previously generated ones**, until the model decides to stop.

In technical terms, the generation ends when the model produces a special token called **<EOS>** (End of Sequence), signaling completion.

Using LLM to solve a task (**Chat Completion**) -Auto-regression



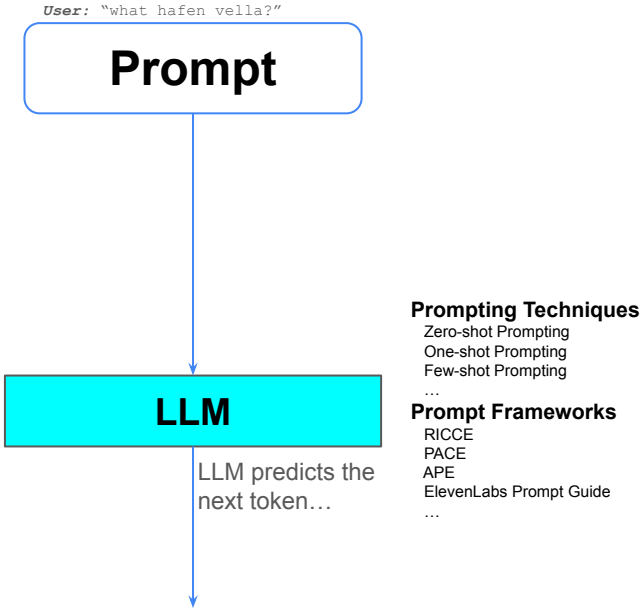
User: "what hafen vella?"
Assistant: "The phrase "What hafen vella?" is a viral meme that originated from a 2013 episode of It's Showtime. <EOS>

This **autoregressive** behavior continues token by token.

Each new token is predicted based on **all previously generated ones**, until the model decides to stop.

In technical terms, the generation ends when the model produces a special token called **<EOS>** (End of Sequence), signaling completion.

Using LLM to solve a task (**Chat Completion**) -Auto-regression

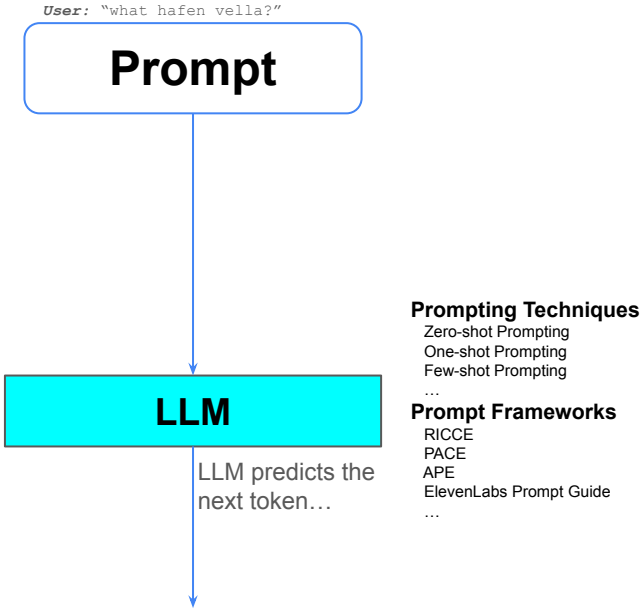


User: "what hafen vella?"
Assistant: "The phrase "What hafen vella?" is a viral meme that originated from a 2013 episode of It's Showtime. <EOS>

While autoregressive LLMs generate text one token at a time based on prior context…

There's one fundamental limitation:

Using LLM to solve a task (**Chat Completion**) -Auto-regression



While autoregressive LLMs generate text one token at a time based on prior context…

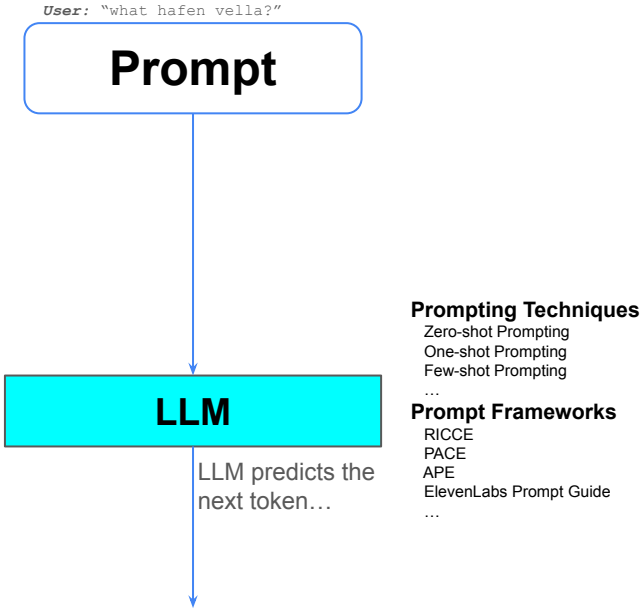
There's one fundamental limitation:

There's no "back" button.

Once a token is generated, the model can't revise it. It can only move forward, even if it realizes something could have been phrased better.

It's **auto-regressive**.
It can only move forward, never backward.

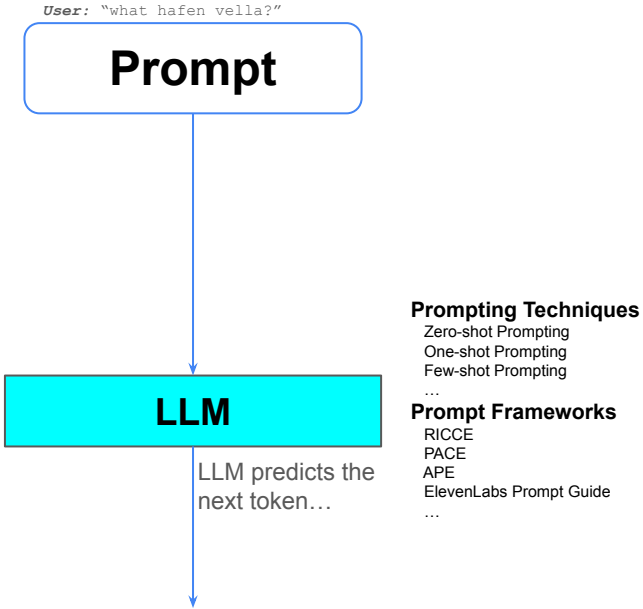
Using LLM to solve a task (**Chat Completion**) -Auto-regression



User: "what hafen vella?"
Assistant: "The phrase "What hafen vella?" is a viral meme that originated from a 2013 episode of It's Showtime. <EOS>

In ChatGPT, what do you usually do if you want to refine its answer?

Using LLM to solve a task (**Chat Completion**) -Auto-regression



User: "what hafen vella?"
Assistant: "The phrase "What hafen vella?" is a viral meme that originated from a 2013 episode of It's Showtime. <EOS>

In ChatGPT, what do you usually do if you want to refine its answer?

You probably just...
✅ Ask a follow-up question
✅ Rephrase your prompt

Nikko Carlo Yabut

Multi-Turn Conversation

Intro to Agentic AI

Dialogue in LLMs

In ChatGPT, you usually ask a follow-up question to get a “better” answer.

This back-and-forth is called a **multi-turn conversation**.

Dialogue in LLMs - Building up the Conversation

```
messages = [ ]
```

Dialogue in LLMs - Building up the Conversation

```
messages = [
    {
        "role": "system",
        "content": "You are a helpful assistant who provides clear, concise answers."
    },
]
```

Dialogue in LLMs - Building up the Conversation


```
messages = [
    {
        "role": "system",
        "content": "You are a helpful assistant who provides clear, concise answers."
    },
    {
        "role": "user",
        "content": "What's a good marketing strategy for a new mobile app?"
    },
]
```

What's a good marketing strategy for a new mobile app?

Dialogue in LLMs - Building up the Conversation


```
messages = [
    {
        "role": "system",
        "content": "You are a helpful assistant who provides clear, concise answers."
    },
    {
        "role": "user",
        "content": "What's a good marketing strategy for a new mobile app?"
    },
    {
        "role": "assistant",
        "content": (
            "A good strategy includes identifying your target audience, "
            "launching with early access or referral incentives, and leveraging "
            "social media and influencer partnerships."
        )
    },
]
```

What's a good marketing strategy for a new mobile app?

A good strategy includes identifying your target audience, launching with early access or referral incentives, and leveraging social media and influencer partnerships.

Dialogue in LLMs - Building up the Conversation


```
messages = [
    {
        "role": "system",
        "content": "You are a helpful assistant who provides clear, concise answers."
    },
    {
        "role": "user",
        "content": "What's a good marketing strategy for a new mobile app?"
    },
    {
        "role": "assistant",
        "content": (
            "A good strategy includes identifying your target audience, "
            "launching with early access or referral incentives, and leveraging "
            "social media and influencer partnerships."
        )
    },
    {
        "role": "user",
        "content": "Can you include examples of referral incentives?"
    },
]
```

What's a good marketing strategy for a new mobile app?

A good strategy includes identifying your target audience, launching with early access or referral incentives, and leveraging social media and influencer partnerships.

Can you include examples of referral incentives?

Dialogue in LLMs - Building up the Conversation


```
messages = [
    {
        "role": "system",
        "content": "You are a helpful assistant who provides clear, concise answers."
    },
    {
        "role": "user",
        "content": "What's a good marketing strategy for a new mobile app?"
    },
    {
        "role": "assistant",
        "content": (
            "A good strategy includes identifying your target audience, "
            "launching with early access or referral incentives, and leveraging "
            "social media and influencer partnerships."
        )
    },
    {
        "role": "user",
        "content": "Can you include examples of referral incentives?"
    },
    {
        "role": "assistant",
        "content": (
            "Sure! Examples include giving users a discount or premium feature "
            "for each friend they refer, or offering both parties in a referral bonus."
        )
    }
]
```

What's a good marketing strategy for a new mobile app?

A good strategy includes identifying your target audience, launching with early access or referral incentives, and leveraging social media and influencer partnerships.

Can you include examples of referral incentives?

Sure! Examples include giving users a discount or premium feature for each friend they refer, or offering both parties in a referral bonus.

Dialogue in LLMs - Building up the Conversation


```
messages = [
    {
        "role": "system",
        "content": "You are a helpful assistant who provides clear, concise answers."
    },
    {
        "role": "user",
        "content": "What's a good marketing strategy for a new mobile app?"
    },
    {
        "role": "assistant",
        "content": (
            "A good strategy includes identifying your target audience, "
            "launching with early access or referral incentives, and leveraging "
            "social media and influencer partnerships."
        )
    },
    {
        "role": "user",
        "content": "Can you include examples of referral incentives?"
    },
    {
        "role": "assistant",
        "content": (
            "Sure! Examples include giving users a discount or premium feature "
            "for each friend they refer, or offering both parties in a referral bonus."
        )
    }
    {
        "role": "user",
        "content": "How can I reach early adopters effectively?"
    },
    {
        "role": "assistant",
        "content": (
            "Engage early adopters through online communities, tech forums, product hunt listings, "
            "and exclusive beta invites to create a sense of exclusivity and momentum."
        )
    },
]
```

Dialogue in LLMs - Building up the Conversation


```
messages = [
    {
        "role": "system",
        "content": "You are a helpful assistant who provides clear, concise answers."
    },
    {
        "role": "user",
        "content": "What's a good marketing strategy for a new mobile app?"
    },
    {
        "role": "assistant",
        "content": (
            "A good strategy includes identifying your target audience, "
            "launching with early access or referral incentives, and leveraging "
            "social media and influencer partnerships."
        )
    },
    {
        "role": "user",
        "content": "Can you include examples of referral incentives?"
    },
    {
        "role": "assistant",
        "content": (
            "Sure! Examples include giving users a discount or premium feature "
            "for each friend they refer, or offering both parties in a referral bonus."
        )
    }
    {
        "role": "user",
        "content": "How can I reach early adopters effectively?"
    },
    {
        "role": "assistant",
        "content": (
            "Engage early adopters through online communities, tech forums, product hunt listings, "
            "and exclusive beta invites to create a sense of exclusivity and momentum."
        )
    },
]
```

Dialogue in LLMs - Building up the Conversation


```
messages = [
    {
        "role": "system",
        "content": "You are a helpful assistant who provides clear, concise answers."
    },
    {
        "role": "user",
        "content": "What's a good marketing strategy for a new mobile app?"
    },
    {
        "role": "assistant",
        "content": (
            "A good strategy includes identifying your target audience, "
            "launching with early access or referral incentives, and leveraging "
            "social media and influencer partnerships."
        )
    },
    {
        "role": "user",
        "content": "Can you include examples of referral incentives?"
    },
    {
        "role": "assistant",
        "content": (
            "Sure! Examples include giving users a discount or premium feature "
            "for each friend they refer, or offering both parties in a referral bonus."
        )
    }
    {
        "role": "user",
        "content": "How can I reach early adopters effectively?"
    },
    {
        "role": "assistant",
        "content": (
            "Engage early adopters through online communities, tech forums, product hunt listings, "
            "and exclusive beta invites to create a sense of exclusivity and momentum."
        )
    },
    {
        "role": "user",
        "content": "Should I focus on one platform first, like iOS?"
    },
    {
        "role": "assistant",
        "content": (
            "Yes, starting with one platform helps you focus your development, gather feedback faster, "
            "and reduce complexity early on—iOS is often chosen due to user spending habits."
        )
    },
]
```

Dialogue in LLMs - Building up the Conversation


```
messages = [
  {
    "role": "system",
    "content": "You are a helpful assistant who provides clear, concise answers."
  },
  {
    "role": "user",
    "content": "What's a good marketing strategy for a new mobile app?"
  },
  {
    "role": "assistant",
    "content": (
      "A good strategy includes identifying your target audience, "
      "launching with early access or referral incentives, and leveraging "
      "social media and influencer partnerships."
    )
  },
  {
    "role": "user",
    "content": "Can you include examples of referral incentives?"
  },
  {
    "role": "assistant",
    "content": (
      "Sure! Examples include giving users a discount or premium feature "
      "for each friend they refer, or offering both parties in a referral bonus."
    )
  },
  {
    "role": "user",
    "content": "How can I reach early adopters effectively?"
  },
  {
    "role": "assistant",
    "content": (
      "Engage early adopters through online communities, tech forums, product hunt listings, "
      "and exclusive beta invites to create a sense of exclusivity and momentum."
    )
  },
  {
    "role": "user",
    "content": "Should I focus on one platform first, like iOS?"
  },
  {
    "role": "assistant",
    "content": (
      "Yes, starting with one platform helps you focus your development, gather feedback faster, "
      "and reduce complexity early on—iOS is often chosen due to user spending habits."
    )
  },
  {
    "role": "user",
    "content": "What metrics should I track after launch?"
  },
  {
    "role": "assistant",
    "content": (
      "Track user retention, daily active users (DAU), customer acquisition cost (CAC), "
      "conversion rates, and app store ratings to measure growth and product-market fit."
    )
  }
]
{
  "role": "user",
  "content": "How can I improve app store visibility?"
},
{
  "role": "assistant",
  "content": (
    "Use App Store Optimization (ASO): include relevant keywords in your title and description, "
    "use high-quality screenshots, and encourage user reviews."
  )
}
},
{
  "role": "user",
  "content": "What kind of content should I post on social media?"
},
{
  "role": "assistant",
  "content": (
    "Post short demos, user testimonials, behind-the-scenes development updates, and "
    "interactive polls to build engagement and interest."
  )
}
},
{
  "role": "user",
  "content": "Is paid advertising worth it early on?"
},
{
  "role": "assistant",
  "content": (
    "Yes, if used strategically. Small, targeted campaigns can help validate messaging, "
    "attract early users, and test different value propositions."
  )
}
},
{
  "role": "user",
  "content": "How often should I update the app after launch?"
},
{
  "role": "assistant",
  "content": (
    "Aim for regular updates—every 2 to 4 weeks—to fix bugs, introduce features, and respond to user feedback."
  )
},
{
  "role": "user",
  "content": "When should I consider expanding to Android?"
},
{
  "role": "assistant",
  "content": (
    "Expand once your iOS version has stable traction and validated product-market fit. "
    "This helps you avoid doubling development efforts too early."
  )
}
]
```


Oops!

We'll soon encounter another problem…

Dialogue in LLMs - Building up the Conversation


```
messages = [
  {
    "role": "system",
    "content": "You are a helpful assistant who provides clear, concise answers."
  },
  {
    "role": "user",
    "content": "What's a good marketing strategy for a new mobile app?"
  },
  {
    "role": "assistant",
    "content": (
      "A good strategy includes identifying your target audience, "
      "launching with early access or referral incentives, and leveraging "
      "social media and influencer partnerships."
    )
  },
  {
    "role": "user",
    "content": "Can you include examples of referral incentives?"
  },
  {
    "role": "assistant",
    "content": (
      "Sure! Examples include giving users a discount or premium feature "
      "for each friend they refer, or offering both parties in a referral bonus."
    )
  },
  {
    "role": "user",
    "content": "How can I reach early adopters effectively?"
  },
  {
    "role": "assistant",
    "content": (
      "Engage early adopters through online communities, tech forums, product hunt listings, "
      "and exclusive beta invites to create a sense of exclusivity and momentum."
    )
  },
  {
    "role": "user",
    "content": "Should I focus on one platform first, like iOS?"
  },
  {
    "role": "assistant",
    "content": (
      "Yes, starting with one platform helps you focus your development, gather feedback faster, "
      "and reduce complexity early on—iOS is often chosen due to user spending habits."
    )
  },
  {
    "role": "user",
    "content": "What metrics should I track after launch?"
  },
  {
    "role": "assistant",
    "content": (
      "Track user retention, daily active users (DAU), customer acquisition cost (CAC), "
      "conversion rates, and app store ratings to measure growth and product-market fit."
    )
  }
  {
  "role": "user",
  "content": "How can I improve app store visibility?"
  }
  {
    "role": "assistant",
    "content": (
      "Use App Store Optimization (ASO): include relevant keywords in your title and description, "
      "use high-quality screenshots, and encourage user reviews."
    )
  }
  {
  "role": "user",
  "content": "What kind of content should I post on social media?"
  }
  {
    "role": "assistant",
    "content": (
      "Post short demos, user testimonials, behind-the-scenes development updates, and "
      "interactive polls to build engagement and interest."
    )
  }
  {
  "role": "user",
  "content": "Is paid advertising worth it early on?"
  }
  {
    "role": "assistant",
    "content": (
      "Yes, if used strategically. Small, targeted campaigns can help validate messaging, "
      "attract early users, and test different value propositions."
    )
  }
  {
  "role": "user",
  "content": "How often should I update the app after launch?"
  }
  {
    "role": "assistant",
    "content": (
      "Aim for regular updates—every 2 to 4 weeks—to fix bugs, introduce features, and respond to user feedback."
    )
  }
  {
    "role": "user",
    "content": "When should I consider expanding to Android?"
  }
  {
    "role": "assistant",
    "content": (
      "Expand once your iOS version has stable traction and validated product-market fit. "
      "This helps you avoid doubling development efforts too early."
    )
  }
]
```

Oops!

We'll soon encounter another problem…

Your context is building up…

As we keep adding turns, we're quietly creating a new challenge, ever-increasing context.

Dialogue in LLMs - Building up the Conversation


```
[
  {
    "role": "system",
    "content": "You are a helpful assistant who provides clear, concise answers."
  },
  {
    "role": "user",
    "content": "What's a good marketing strategy for a new mobile app?"
  },
  {
    "role": "assistant",
    "content": (
      "A good strategy includes identifying your target audience, "
      "launching with early access or referral incentives, and leveraging "
      "social media and influencer partnerships."
    )
  },
  {
    "role": "user",
    "content": "Can you include examples of referral incentives?"
  },
  {
    "role": "assistant",
    "content": (
      "Sure! Examples include giving users a discount or premium feature "
      "for each friend they refer, or offering both parties in a referral bonus."
    )
  },
  {
    "role": "user",
    "content": "How can I reach early adopters effectively?"
  },
  {
    "role": "assistant",
    "content": (
      "Engage early adopters through online communities, tech forums, product hunt listings, "
      "and exclusive beta invites to create a sense of exclusivity and momentum."
    )
  },
  {
    "role": "user",
    "content": "Should I focus on one platform first, like iOS?"
  },
  {
    "role": "assistant",
    "content": (
      "Yes, starting with one platform helps you focus your development, gather feedback faster, "
      "and reduce complexity early on—iOS is often chosen due to user spending habits."
    )
  },
  {
    "role": "user",
    "content": "What metrics should I track after launch?"
  },
  {
    "role": "assistant",
    "content": (
      "Track user retention, daily active users (DAU), customer acquisition cost (CAC), "
      "conversion rates, and app store ratings to measure growth and product-market fit."
    )
  },
  {
    "role": "user",
    "content": "How can I improve app store visibility?"
  },
  {
    "role": "assistant",
    "content": (
      "Use App Store Optimization (ASO): include relevant keywords in your title and description, "
      "use high-quality screenshots, and encourage user reviews."
    )
  },
  {
    "role": "user",
    "content": "What kind of content should I post on social media?"
  },
  {
    "role": "assistant",
    "content": (
      "Post short demos, user testimonials, behind-the-scenes development updates, and "
      "interactive polls to build engagement and interest."
    )
  },
  {
    "role": "user",
    "content": "Is paid advertising worth it early on?"
  },
  {
    "role": "assistant",
    "content": (
      "Yes, if used strategically. Small, targeted campaigns can help validate messaging, "
      "attract early users, and test different value propositions."
    )
  },
  {
    "role": "user",
    "content": "How often should I update the app after launch?"
  },
  {
    "role": "assistant",
    "content": (
      "Aim for regular updates—every 2 to 4 weeks—to fix bugs, introduce features, and respond to user feedback."
    )
  },
  {
    "role": "user",
    "content": "When should I consider expanding to Android?"
  },
  {
    "role": "assistant",
    "content": (
      "Expand once your iOS version has stable traction and validated product-market fit. "
      "This helps you avoid doubling development efforts too early."
    )
  }
]
```


Oops!

We'll soon encounter another problem…

Your context building up…

As we keep adding turns, we're quietly creating a new challenge, ever-increasing context.

Your context might hit the max allowable input, aka context window.

Dialogue in LLMs - **Context Window** Problem

Prompt

```
messages = [
  {
    "role": "system",
    "content": "You are a helpful assistant who provides clear, concise answers."
  },
  {
    "role": "user",
    "content": "What's a good marketing strategy for a new mobile app?"
  },
  {
    "role": "assistant",
    "content": (
      "A good strategy includes identifying your target audience, "
      "launching with early access or referral incentives, and leveraging "
      "social media and influencer partnerships."
    )
  },
  {
    "role": "user",
    "content": "Can you include examples of referral incentives?"
  },
  {
    "role": "assistant",
    "content": (
      "Sure! Examples include giving users a discount or premium feature "
      "for each friend they refer, or offering both parties in a referral bonus."
    )
  },
  {
    "role": "user",
    "content": "How can I reach early adopters effectively?"
  },
  {
    "role": "assistant",
    "content": (
      "Engage early adopters through online communities, tech forums, product hunt listings, "
      "and exclusive beta invites to create a sense of exclusivity and momentum."
    )
  },
  {
    "role": "user",
    "content": "Should I focus on one platform first, like iOS?"
  },
  {
    "role": "assistant",
    "content": (
      "Yes, starting with one platform helps you focus your development, gather feedback faster, "
      "and reduce complexity early on—iOS is often chosen due to user spending habits."
    )
  },
  {
    "role": "user",
    "content": "What metrics should I track after launch?"
  },
  {
    "role": "assistant",
    "content": (
      "Track user retention, daily active users (DAU), customer acquisition cost (CAC), "
      "conversion rates, and app store ratings to measure growth and product-market fit."
    )
  },
{
  "role": "user",
  "content": "How can I improve app store visibility?"
},
{
  "role": "assistant",
  "content": (
    "Use App Store Optimization (ASO): include relevant keywords in your title and description, "
    "use high-quality screenshots, and encourage user reviews."
  )
},
{
  "role": "user",
  "content": "What kind of content should I post on social media?"
},
{
  "role": "assistant",
  "content": (
    "Post short demos, user testimonials, behind-the-scenes development updates, and "
    "interactive polls to build engagement and interest."
  )
},
{
  "role": "user",
  "content": "Is paid advertising worth it early on?"
},
{
  "role": "assistant",
  "content": (
    "Yes, if used strategically. Small, targeted campaigns can help validate messaging, "
    "attract early users, and test different value propositions."
  )
},
{
  "role": "user",
  "content": "How often should I update the app after launch?"
},
{
  "role": "assistant",
  "content": (
    "Aim for regular updates—every 2 to 4 weeks—to fix bugs, introduce features, and respond to user feedback."
  )
},
{
  "role": "user",
  "content": "When should I consider expanding to Android?"
},
{
  "role": "assistant",
  "content": (
    "Expand once your iOS version has stable traction and validated product-market fit. "
    "This helps you avoid doubling development efforts too early."
  )
}
]
```

Oops!

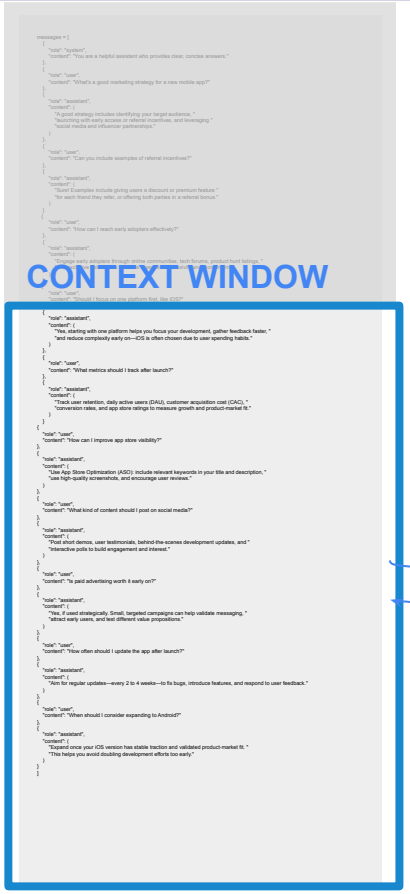
We'll soon encounter another problem…

Your context building up…

As we keep adding turns, we're quietly creating a new challenge, ever-increasing context.

Your context might hit the max allowable input, aka context window.

Dialogue in LLMs - **Context Window** Problem



Oops!

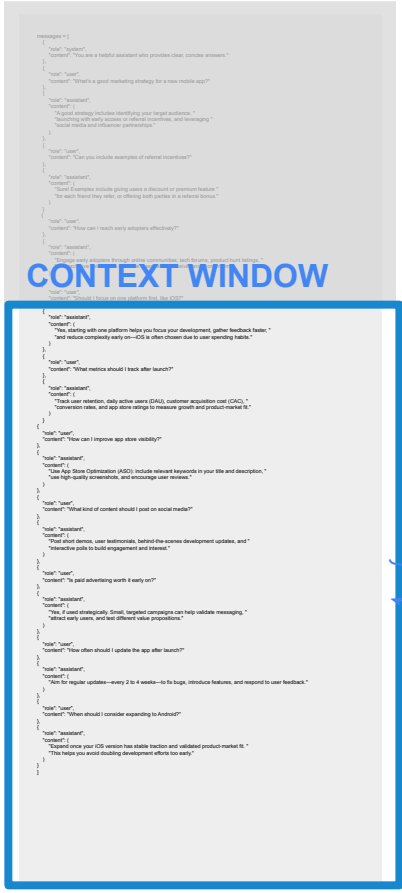
We'll soon encounter another problem…

Your context building up…

As we keep adding turns, we're quietly creating a new challenge, ever-increasing context.

Your context might hit the max allowable input, aka context window.

Dialogue in LLMs - **Context Window** Problem



Year	Model(s)	Context Window	≈ Word Count	≈ Page Count
2020	**GPT-3**	2,049 tokens	~1,537 words	~3 pages
2021	**GPT-3**, **LLaMA 1**, BERT variants	2k–4k tokens	~1,500–3,000 words	~3–6 pages
2022	GPT-3.5, Claude 1, PaLM	4k–8k tokens	~3,000–6,000 words	~6–12 pages

We'll soon encounter another problem…

Your context building up…

As we keep adding turns, we're quietly creating a new challenge, ever-increasing context.

Your context might hit the max allowable input, aka context window.

Dialogue in LLMs - From **Context Window** Problem

📊 LLM Context Window Comparison

Model	Tokens	≈ Words	≈ Pages
GPT-4-turbo (OpenAI)	128,000	~96,000 words	~192 pages
GPT-4 (original)	8,192–32,768	~6,100–24,600	~12–49 pages
GPT-3.5-turbo	16,385	~12,289 words	~25 pages
Claude 3 Opus	200,000	~150,000 words	~300 pages
Gemini 1.5 Pro	1,000,000	~750,000 words	~1,500 pages
Mistral Large	32,000	~24,000 words	~48 pages
Mixtral (MoE)	32,000	~24,000 words	~48 pages
LLaMA 2	4k–32k	~3k–24k words	~6–48 pages
Command R+ (Cohere)	128,000	~96,000 words	~192 pages
Yi-1.5 (01.AI)	32,000	~24,000 words	~48 pages
Grok (xAI)	~128,000	~96,000 words	~192 pages

LLM

```
response = client.chat.completions.create(
    model="gpt-4",
    messages=messages,
    temperature=0.7
)
```

auto regression

💡 **Modern LLMs have *huge* context windows.**

Models like GPT-4-turbo can handle up to **128,000 tokens**, and Claude 3 can go up to **200,000 tokens** — that's the length of a novel or an entire codebase.

So... problem solved?

Dialogue in LLMs - From **Context Window** Problem

📊 LLM Context Window Comparison

Model	Tokens	≈ Words	≈ Pages
GPT-4-turbo (OpenAI)	128,000	~96,000 words	~192 pages
GPT-4 (original)	8,192–32,768	~6,100–24,600	~12–49 pages
GPT-3.5-turbo	16,385	~12,289 words	~25 pages
Claude 3 Opus	200,000	~150,000 words	~300 pages
Gemini 1.5 Pro	1,000,000	~750,000 words	~1,500 pages
Mistral Large	32,000	~24,000 words	~48 pages
Mixtral (MoE)	32,000	~24,000 words	~48 pages
LLaMA 2	4k–32k	~3k–24k words	~6–48 pages
Command R+ (Cohere)	128,000	~96,000 words	~192 pages
Yi-1.5 (01.AI)	32,000	~24,000 words	~48 pages
Grok (xAI)	~128,000	~96,000 words	~192 pages

LLM

```
response = client.chat.completions.create(
    model="gpt-4",
    messages=messages,
    temperature=0.7
)
```

auto regression

💡 **Modern LLMs have *huge* context windows.**

Models like GPT-4-turbo can handle up to **128,000 tokens**, and Claude 3 can go up to **200,000 tokens** — that's the length of a novel or an entire codebase.

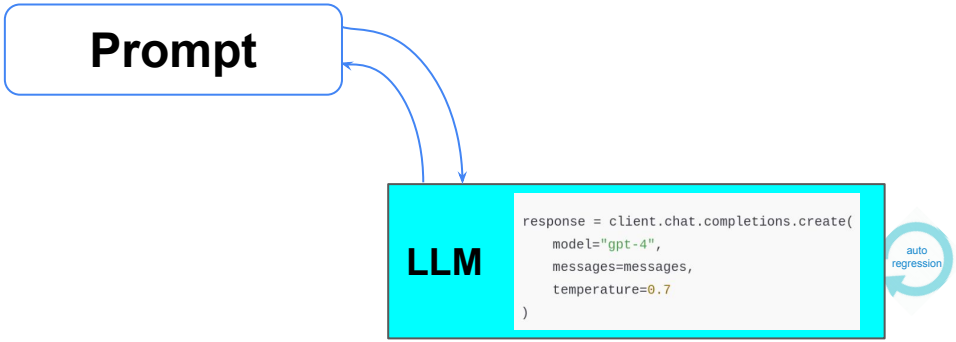
So... problem solved? **Not quite.**

Dialogue in LLMs - From Context Window to **Context QUALITY** Problem

⚠️ But… Now the Problem Is Context Quality

With more room comes new challenges:

Problem	Description
🎯 **Irrelevant or noisy context**	Long input ≠ useful input. The model might attend to wrong things.
🧂 **Low signal-to-noise ratio**	Critical information can get buried under fluff.
🧠 **Lack of structure or salience**	Without cues (headings, bullets, instructions), the model might misunderstand or ignore key parts.
🧮 **Token bloat**	Just because you *can* fit 100k tokens doesn't mean you *should*. It increases latency, cost, and sometimes worsens performance.



💡 Modern LLMs have *huge* context windows.

Models like GPT-4-turbo can handle up to **128,000 tokens**, and Claude 3 can go up to **200,000 tokens** — that's the length of a novel or an entire codebase.

So... problem solved? **Not quite.**

Now that we *can* stuff all that information in, a new challenge emerges:

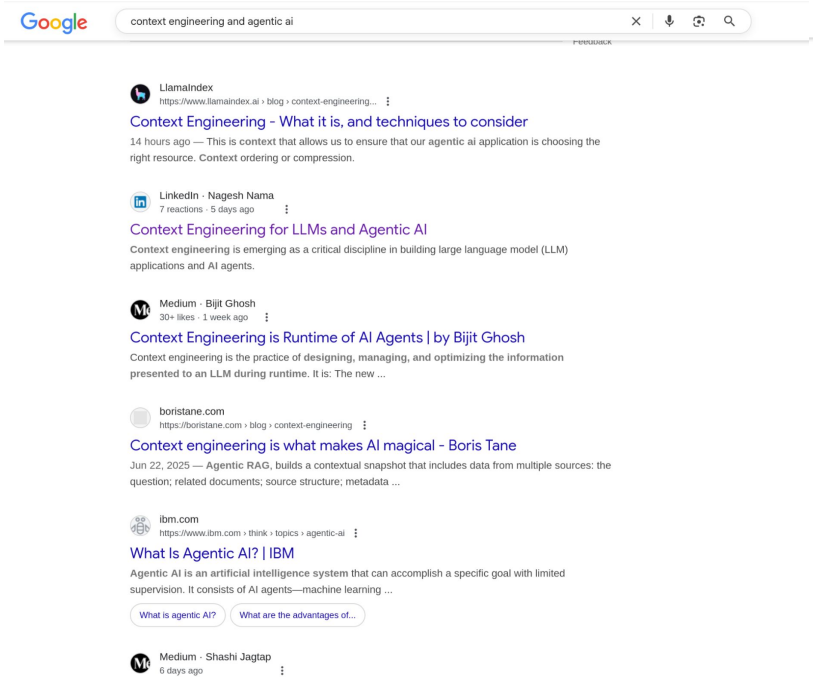
> 🧠 The issue is no longer context *size* — it's context **quality**.

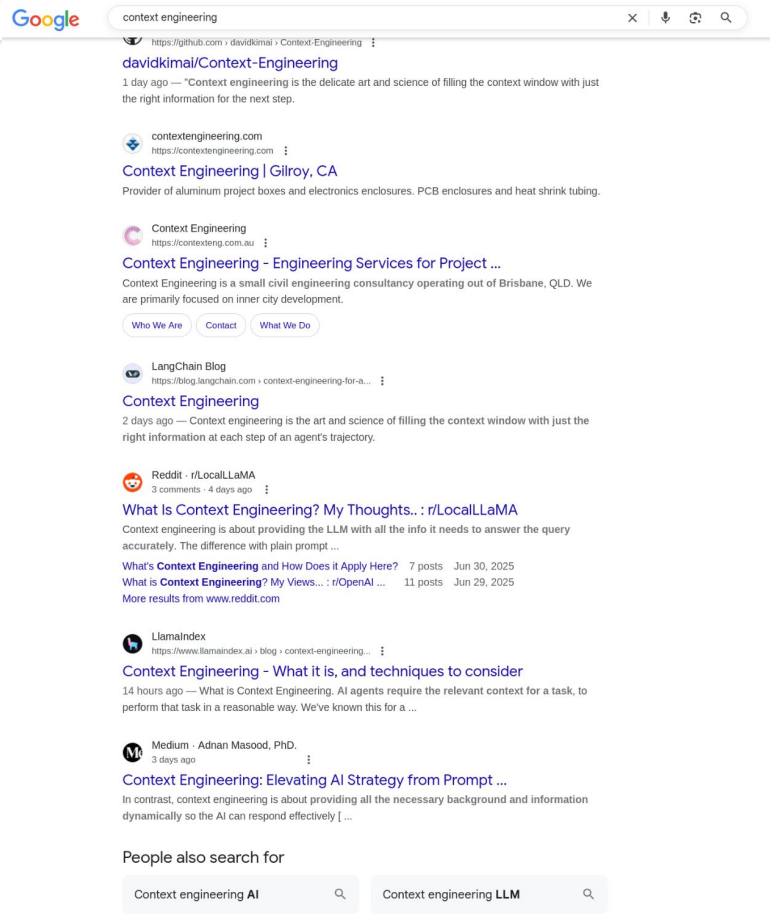
Nikko Carlo Yabut

Context Engineering

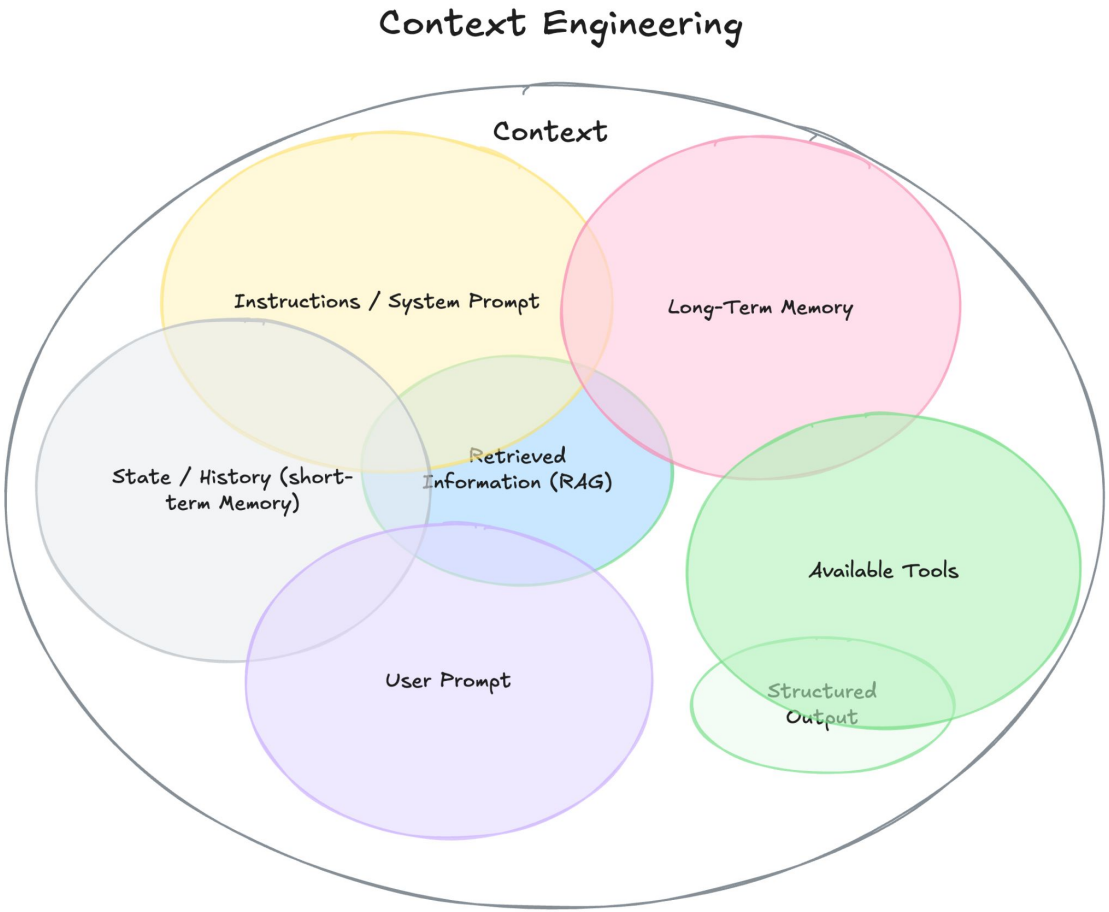
Intro to Agentic AI

Context Engineering





Context Engineering

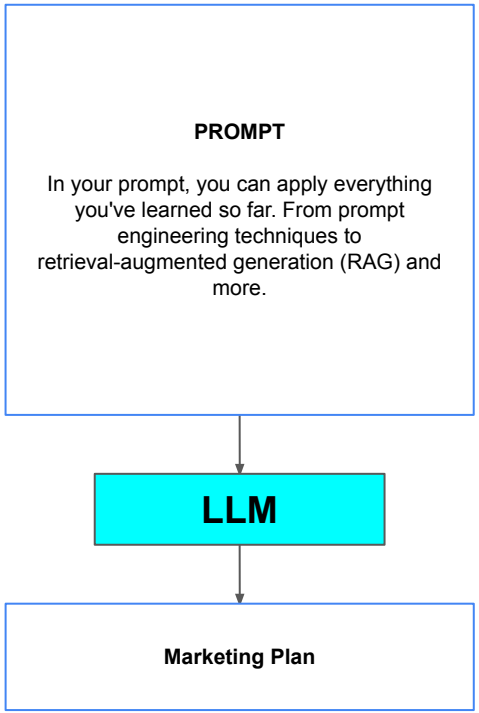


Nikko Carlo Yabut

Compounding LLMs

Intro to Agentic AI

Task: Create a Marketing Plan

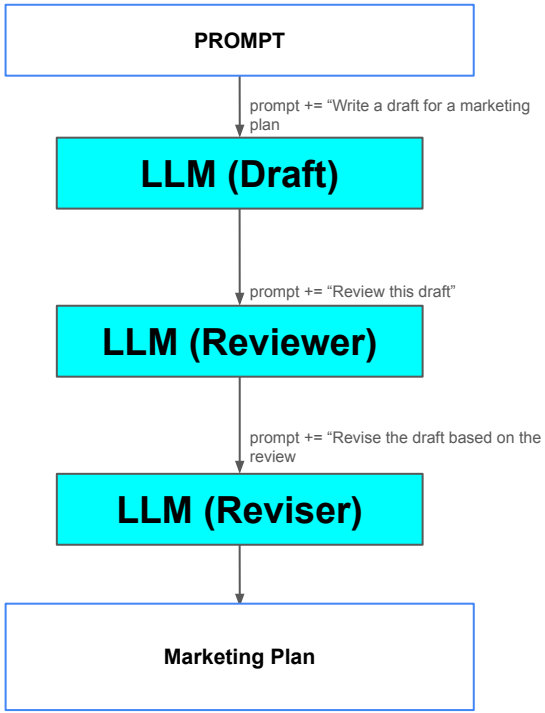


IRL

In real life, we don't just write something once and call it done.
There's usually a **reviewer**, followed by a **revision**, until we arrive at a final version.

Compounding LLM Approach

Task: Create a Marketing Plan

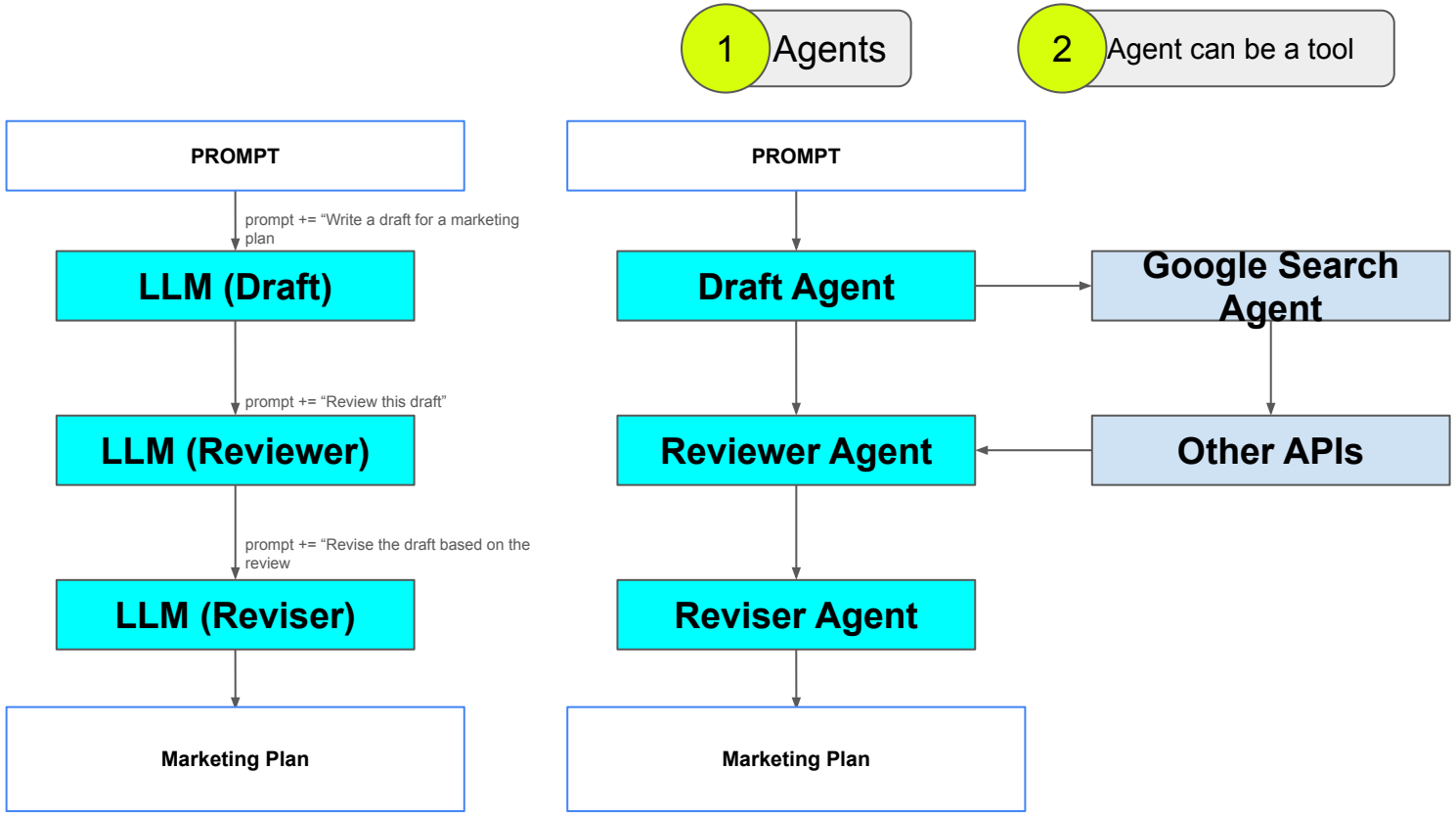


IRL

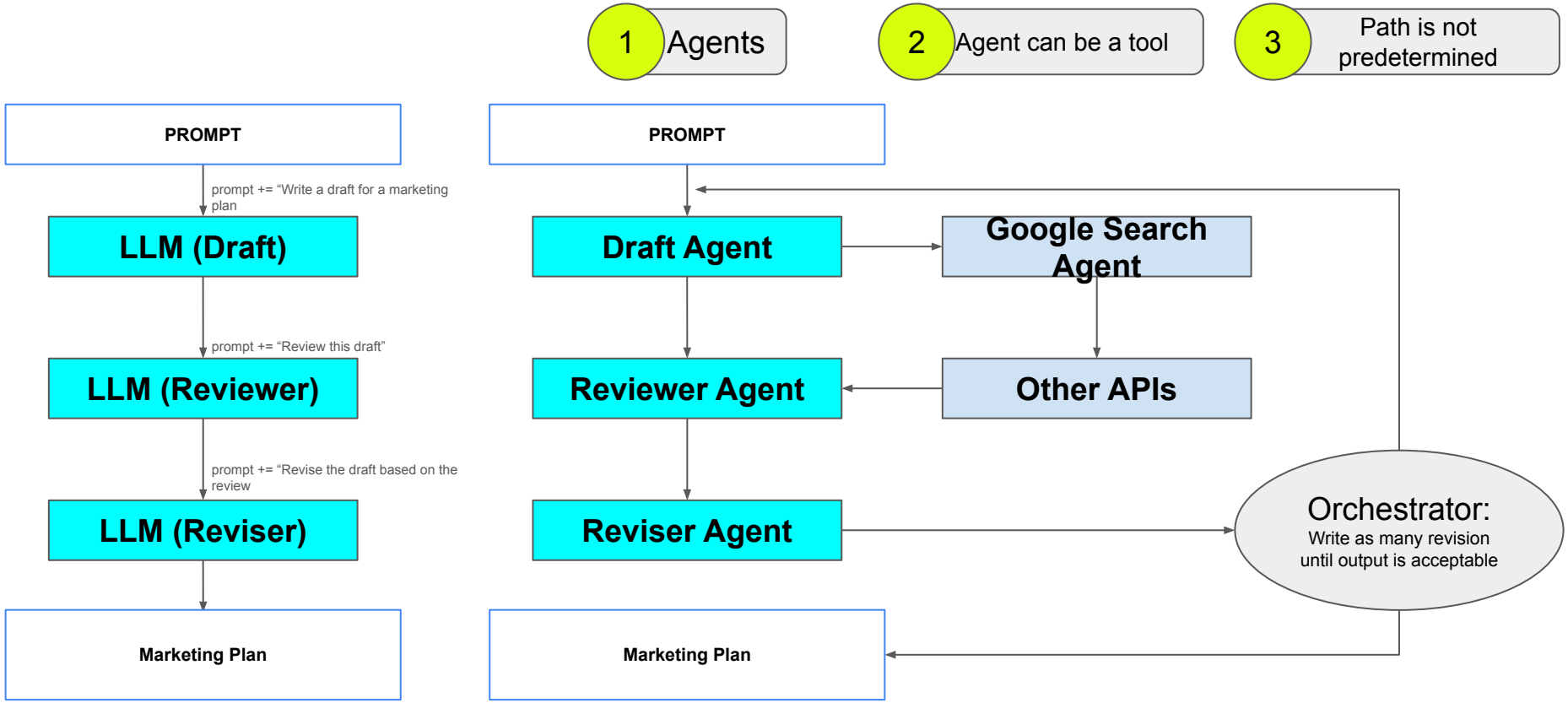
In real life, we don’t just write something once and call it done.
There’s usually a **reviewer**, followed by a **revision**, until we arrive at a final version.

As you can see, this **compounding LLM approach** often yields better results.
your initial draft is **reviewed**, **critiqued**, and then **refined** — just like in real collaborative work.

Compounding LLM Approach as an **Agentic AI**



Compounding LLM Approach as an **Agentic AI**



`[Why go through all this?]`

Task Adaptability

To deliver better results.
By giving each LLM a specific role, we improve the quality of its context — and in turn, the quality of its output.

Other examples

1. 🧑‍⚖️ Legal Document Review Agent System

Task: Draft and finalize a legal contract.

Agentic Workflow:

- 📝 **Drafting Agent:** Writes the initial version of a contract.
- 🔍 **Compliance Agent:** Reviews for legal inconsistencies, missing clauses, or regulatory gaps.
- ⚖️ **Risk Assessment Agent:** Flags liability risks or vague terms.
- ✅ **Finalization Agent:** Decides if the contract is ready for use or needs revision.

Why it's agentic:
Each agent has a **defined responsibility**, and the output **loops** until the contract passes the legal bar set by the final decision agent.

2. 🧪 Research Paper Generator

Task: Generate a literature review or technical research summary.

Agentic Workflow:

- 🔎 **Retriever Agent:** Uses RAG to fetch relevant academic papers.
- 🧠 **Summarizer Agent:** Extracts key findings and contributions.
- 🧰 **Critique Agent:** Reviews the summary for gaps, outdated studies, or bias.
- 🛠️ **Refinement Agent:** Improves coherence, flow, and adds missing references.
- 🗒️ **Final Check Agent:** Assesses whether the paper is complete and publish-ready.

Why it's agentic:
This chain transforms **raw data** into a refined academic artifact through **role-specific reasoning**, with **multi-turn refinement**.

Other examples

3. 🛍️ AI Personal Shopper Agent System

Task: Help a user find the perfect outfit for an event based on preferences, budget, and availability.

Agentic Workflow:

- 👤 **User Profiling Agent:** Gathers user style, size, color preferences, and budget.
- 🔍 **Trend Scanner Agent:** Analyzes current fashion trends and seasonal styles.
- 🛒 **Inventory Scout Agent:** Searches online stores for matching items in stock.
- 🤝 **Negotiator Agent:** Applies discount codes or selects optimal bundles.
- 🧠 **Decision Agent:** Presents 3 curated outfit options, justifying the selection.

Why it's agentic:
Each agent **acts autonomously** on a **different reasoning layer** (style, availability, price), then composes a practical, real-world action outcome — a shopping decision.

4. 🧭 Disaster Response Planning Agent Team

Task: Generate a real-time action plan in response to a natural disaster alert.

Agentic Workflow:

- 🌍 **Geolocation Agent:** Identifies affected regions and population density using real-time maps.
- 📦 **Resource Allocation Agent:** Matches available emergency supplies and personnel to locations in need.
- 🔗 **Communication Agent:** Drafts and sends out alerts to local officials and the public.
- ⏱️ **Time Optimization Agent:** Prioritizes response routes and supply chains based on urgency and accessibility.
- ✅ **Supervisor Agent:** Validates the plan and triggers autonomous deployment or asks for human approval.

Why it's agentic:
This system **acts in the physical world**, not just with words — agents gather, reason, plan, and act, combining LLMs with geospatial tools and real-world data streams.

Other examples

5. 🎬 Video Editor Agent Team (for YouTubers or Content Creators)

Task: Turn a raw video recording into an engaging, publish-ready YouTube video.

Agentic Workflow:

- 🧠 **Script Analyzer Agent:** Parses spoken audio or transcript to understand the story structure and key highlights.
- ✂️ **Editing Agent:** Cuts silences, filler words, or irrelevant segments; applies pacing adjustments.
- 🎨 **Visual Enhancer Agent:** Adds dynamic text overlays, transitions, zoom effects, or B-roll suggestions.
- 🎵 **Audio Cleaner Agent:** Reduces background noise, balances volume, and inserts background music based on tone.
- 🧾 **Summary Agent:** Generates video title, description, tags, and thumbnail caption based on final content.

Why it's agentic:
The agents **collaborate asynchronously** to refine different aspects of the video, and their decisions are **context-aware** — e.g., matching visuals to speech content.

6. 🎧 Real-Time Podcast Producer Agent

Task: Assist live podcast recording with enhancements and automation.

Agentic Workflow:

- 🗣️ **Speech-to-Text Agent:** Transcribes live conversation into text in real-time.
- 🧠 **Insight Agent:** Highlights memorable quotes, topic shifts, and emotional tone changes.
- 🔊 **Audio Engineering Agent:** Automatically adjusts levels, removes mic pops, and applies compression as you speak.
- 🎙️ **Fact-Checking Agent:** Flags inaccurate claims or prompts the host with missing info during breaks.
- 📝 **Post-Production Agent:** Generates show notes, timestamps, social media snippets, and a title.

Why it's agentic:
This system doesn't just process — it **listens, thinks, and prepares output** across modalities (speech, text, metadata), enhancing both live quality and downstream publishing.

Demo and Activity

https://colab.research.google.com/drive/1pmy5U2TnEegQ_m_sWbmXALmhu3tazSc6#scrollTo=NieN8Fe7Oi0Y

Business Case Ideation

Build an Agentic Business Case

1. Define the problem

   What's the core issue?
   Who experiences it?

2. Who is your user?

   Who feels this problem?
   Ex: *Support reps*, *Students*, *HR staff*, etc.

3. Design the Agent Team

   Create 3–5 agents with distinct roles.
   For each agent, describe:

- Name and function
- What modality it works with (text, audio, video, etc.)
  What value it adds to the overall workflow

4. Data Flow - Why is it Agentic?

Map-out the whole process

```
Lecture Audio/Video
    --> Speech-to-Text Agent
        --> Transcript
            --> Topic Segmentation Agent
                --> Segmented Transcript (by topic)
                    --> Insight Extraction Agent
                        --> Key Points, Definitions, Takeaways
                            --> Visual Slide Generator Agent
                                --> Auto-Generated Slides
                            --> Summary & Quiz Agent
                                --> 1-Page Summary + Quiz
```