Large Language Models

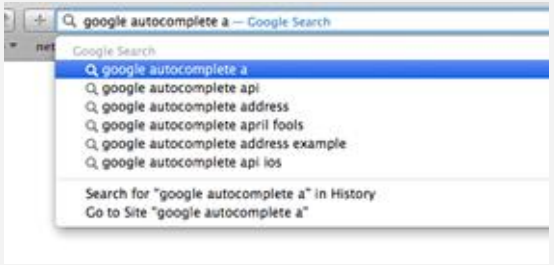
The Core Task

- **Predict the next word given the previous words**

User: ”The capital of France is ”

Model predicts:

Paris	0.82
London	0.05
Rome	0.04
city	0.03



Tokens instead of Words

- LLMs do not operate directly with words. They operate with **tokens.**

Input: ”ChatGPT is amazing”

Tokens: (Chat, GPT, is, amaz, ing)

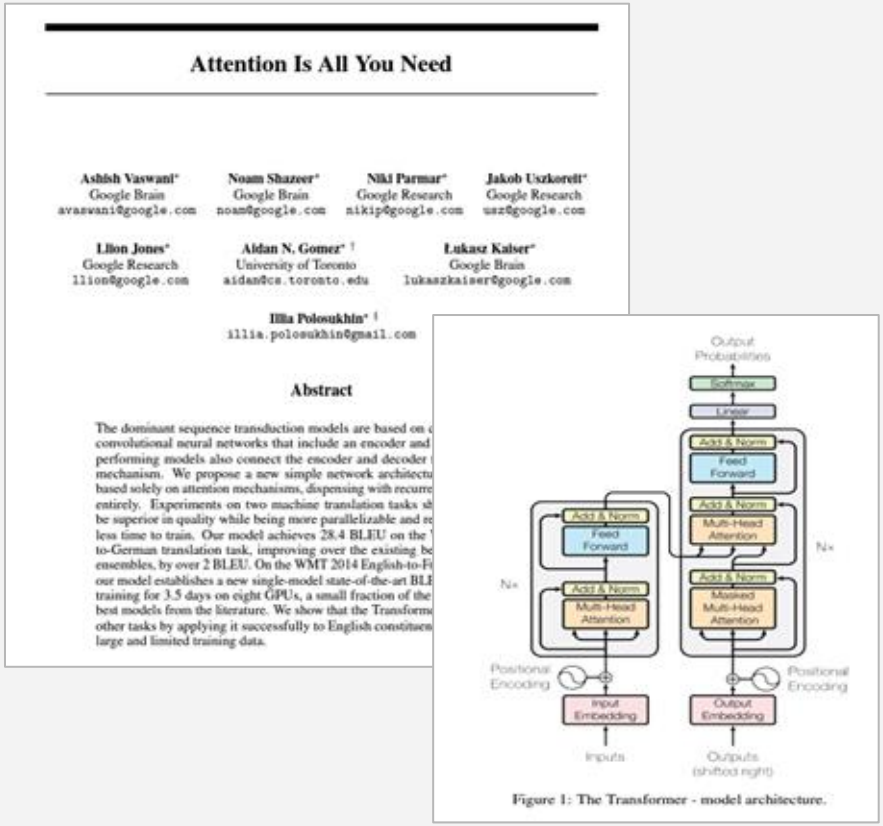
Language Patterns

- Grammer: `"I am going to the ___" → store / park / office`
- Facts: `"The capital of France is ___" → Paris`
- Reasoning: `"It rained all night, so the ground is ___" → wet`
- Programming patterns

```
for i in range(10):
   print(i)
```

- Writing styles: `"Once upon a time in a distant ___" → kingdom`

2017: The Breakthrough That Changed AI



Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

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

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* † illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on c convolutional neural networks that include an encoder and performing models also connect the encoder and decoder mechanism. We propose a new simple network architectu based solely on attention mechanisms, dispensing with recurre entirely. Experiments on two machine translation tasks sh be superior in quality while being more parallelizable and re less time to train. Our model achieves 28.4 BLEU on the to-German translation task, improving over the existing be ensembles, by over 2 BLEU. On the WMT 2014 English-to-F our model establishes a new single-model state-of-the-art BLE training for 3.5 days on eight GPUs, a small fraction of the best models from the literature. We show that the Transform other tasks by applying it successfully to English constituen large and limited training data.

Figure 1: The Transformer - model architecture.

"Attention is All You Need"

From Sequential Reading -> Attention

- **Before (RNNs)**
  - Reads text **one word at a time**
  - Struggles with long context

- **After (Transformers)**
  - Uses **attention**
  - Looks at the **whole sentence** and connects relevant words

- **Example:** “The cat sat on the mat because it was tired.”
  - To understand **what “it” means**, you need to look back and decide what word matters most. “it” -> “**cat**”

What is “Attention”?

- **Attention = focusing on what matters**
  - For each word, the model asks**: Which other words should I pay attention to right now?**

- **Transformer** = attention-based model
  - Built around attention
  - Understands context dynamically

Base LLM Behavior

- A **base LLM** is only trained to predict the next token.
- It **does not know it is supposed to answer questions.**

Input: ”Once upon a time, there was a unicorn”
Response: ”that lives in a magical forest”

Input: “What is the capital of France?”
Response: “What is France’s population?”

Instruction-Tuned LLMs

- To make models behave like assistants, they undergo **instruction tuning.**

Instruction: ”What is the capital of France?”
Response: ”The capital of France is Paris.”

- This teaches the model to:
  - Answer questions
  - Follow instructions
  - Behave helpfully

Human Feedback (RLHF)

- Humans compare model responses and choose the better one.

Instruction: ”Explain black holes”

Response A

“Black holes are regions of space where gravity is so strong that nothing, not even light, can escape.”

Response B

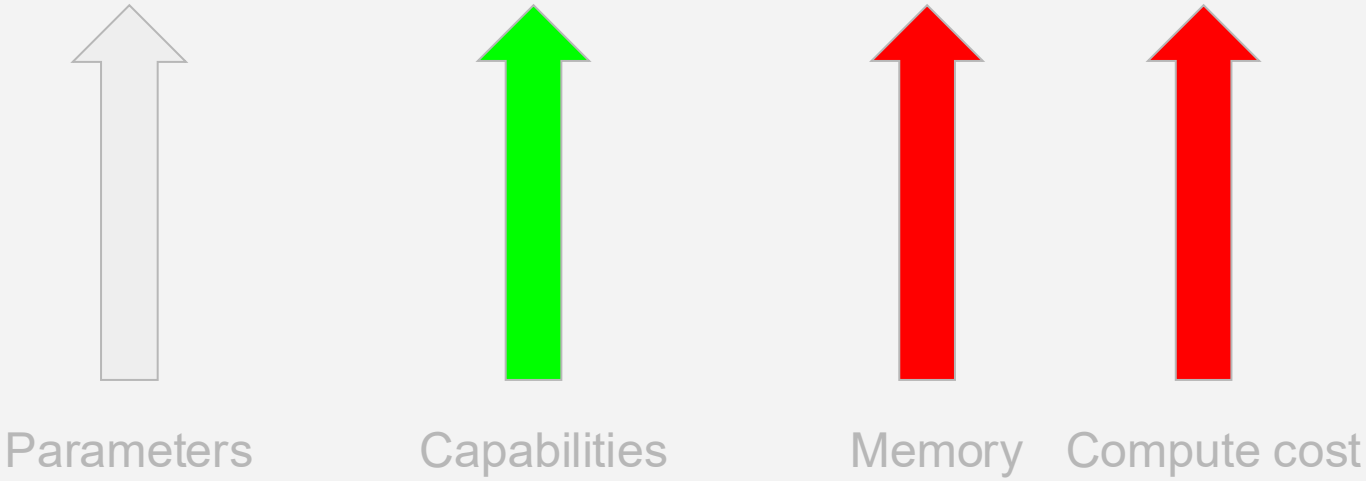
“Black holes are weird space objects that eat things.”

Probability, Not Understanding

- **The Illusion of Thought** - It looks like reasoning, but it is math.
- **Probabilistic Engine** - It calculates the most likely next word based on context.
- **The AI does not "know" facts** - it knows which words tend to appear near each other.

Number of Parameters

LLM parameters are the model's learned weights - the "capacity" that lets it understand and generate language.



Where Can LLM Run?

7B–14B models can run on personal computers with consumer GPUs (or even CPU with quantization)

Personal computer

30B+ models typically require high-memory data-center GPUs (A100/H100) or multi-GPU setups

Cloud

Some Popular LLMs

Closed LLMs

Model Name	Company	Key Features
GPT-5.4	OpenAI	Top "All-Rounder", Advanced Agents
Claude 4.6 Opus	Anthropic	Best Coding & Nuance, Low Hallucination
Gemini 3	Google	Real-time Video/Audio, 2M+ Context
o3 (Reasoning)	OpenAI	Deep Logic (Ph.D. level Math/Science)
Grok 3	xAI	Uncensored, Real-time X (Twitter) Data

Open LLMs

Model Name	Company	Parameters	Key Features
Llama 4 Scout	Meta	10B, 85B, 500B	Massive 10M Context, Enterprise Standard
Qwen 3	Alibaba	7B, 32B, 110B	Strong Math & Vision, Efficient
DeepSeek-V3.2	DeepSeek	280B (MoE)	Best Value, Top-Tier Coding
Phi 4	Microsoft	4B, 16B	Best Small Model (Runs on Laptop)

Improving LLM Performance

Adaption the Foundation Model

Input: *Patient presents with severe headaches and nausea….*
Output: *Summary: Patient shows symptoms consistent with migraine.*

Full Fine Tuning
Retrain the base model by updating 100% the model's parameters (weights)

Parameter Efficient Tuning
Retrain the base model by updating only a small subset of parameters!

Prompt Engineering

Improve model outputs by designing better prompts.

Prompt: `Summarize this document.`

Better prompt: `Summarize the document in three bullet points focusing on key risks.`

Prompt and Context Engineering

The Principles

1. Be Specific About the Task

Why it matters:
Vague instructions force AI to guess what you want. The more specific you are about the task, the less guesswork involved - and the more useful the result.

❌ Bad
“Help me with this report.”

✅ Good
“Review the executive summary of this report and suggest how to make it clearer for a non-technical audience.”

2. Provide Context

Why it matters:
AI doesn't know your situation unless you explain it. Context includes background information, who the audience is, what you've already tried, and any constraints.

❌ Bad
"Write an email about the deadline change."

✅ Good
"Write an email to our client (a small law firm we've worked with for 2 years) explaining that the project deadline is moving from March 15 to April 1 due to a delay in receiving their input documents. Keep it professional but warm."

3. Define the Output Format

Why it matters:
Telling AI what format you need saves time and ensures the output fits your workflow. Do you want bullet points, a table, a paragraph, a numbered list? Say so.

❌ Bad
“What are the key points from these meeting notes?”

✅ Good
“Extract the action items from these meeting notes. List each one with the owner’s name and due date in a table.”

4. Assign a Role or Perspective

Why it matters:
Giving AI a role helps it adopt the right tone, depth, and expertise level. “Act as a financial analyst” produces different output than “act as a customer support agent.”

❌ Bad
“Give me feedback on my presentation.”

✅ Good
“Act as a skeptical senior executive who has limited time. Review my presentation and point out where I might lose their attention or where my argument is weak.”

5. Break Down Complex Requests

Why it matters:
Large, multi-part requests often produce incomplete or unfocused results. Breaking a task into steps gives you more control and better quality at each stage.

❌ Bad
“Analyze our Q3 sales data and create a full quarterly review presentation.”

✅ Good approach
Start with “Summarize the main trends in this Q3 sales data.” Then: “Identify any regions or products that underperformed compared to Q2.” Then: “Draft an outline for a 10-minute presentation covering these findings.”

6. Provide Examples

Why it matters:
Showing AI what good output looks like is one of the most effective ways to get what you want. Examples communicate style, format, and quality standards more clearly than descriptions alone.

❌ Bad prompt
“Write product descriptions for my online store.”

✅ Good prompt
“Write product descriptions for my online store. Here’s an example of the style I want: ‘The Horizon Backpack combines rugged durability with minimalist design. Water-resistant canvas, padded laptop sleeve, lifetime warranty. Perfect for daily commutes or weekend adventures.’ Now write similar descriptions for these three products…”

7. Iterate and Refine

Why it matters:
Your first prompt is a starting point, not the finish line. When the output isn't quite right, apply these same principles to your feedback. Don't just say "try again" - be specific about what to change.

❌ Bad feedback
"This isn't what I wanted. Make it better."

✅ Good feedback
"This is too formal for our team culture. Make the tone more conversational, and shorten the introduction - we can skip the background since everyone already knows the project."

Prompt and Context Engineering - Summary

1. Be Specific About the Task
2. Provide Context
3. Define the Output Format
4. Assign a Role or Perspective
5. Break Down Complex Requests
6. Provide Examples
7. Iterate and Refine

LLM Based System

Chat Prompts

LLM chat requests are structured as messages.

System: ”You are a helpful financial assistant.”
User: ”Summarize this report.”
Assistant: ”Here is a summary...”

Chat History

- **LLMs do not store memory between requests.**

- Every message must include the conversation history again.

```
System: ”You are a helpful assistant.”
User: ”What is the capital of France?”
Assistant: ”The capital of France is Paris.”

User: What is its population?
Assistant: ”The Population of it is ...”
```

Memory

Short-Term Memory

Contextual Understanding

Recent Context

Long-Term Memory

(Retrieval)

Stored Knowledge

RAG

Retrieval-Augmented Generation

Long Term Memory

Long-term memory allows an AI system to remember beyond a single conversation.

Used to:

- Persist information across sessions
- Access external knowledge sources
- Retrieve relevant data when needed

Full Context vs. Retrieval (RAG)

Full Context

Everything Included

Retrieval Methods

Selective Retrieval

How Does RAG Work?

1. User asks a question
2. System searches external knowledge
3. Relevant information is added to the prompt
4. LLM generates an answer using that context

Distributional Hypothesis

"A word is characterized by the company it keeps"

"The **dolphin** leaped out of the waves."
"We watched a **dolphin** at the aquarium."
"The **dolphin** swam alongside the boat."

"She walked into the **bank** and deposited her paycheck."
"She sat on the **bank** of the river and watched the water flow."

Word Embeddings

- Words are represented as points in a high-dimensional space.
- Similar words appear closer together.

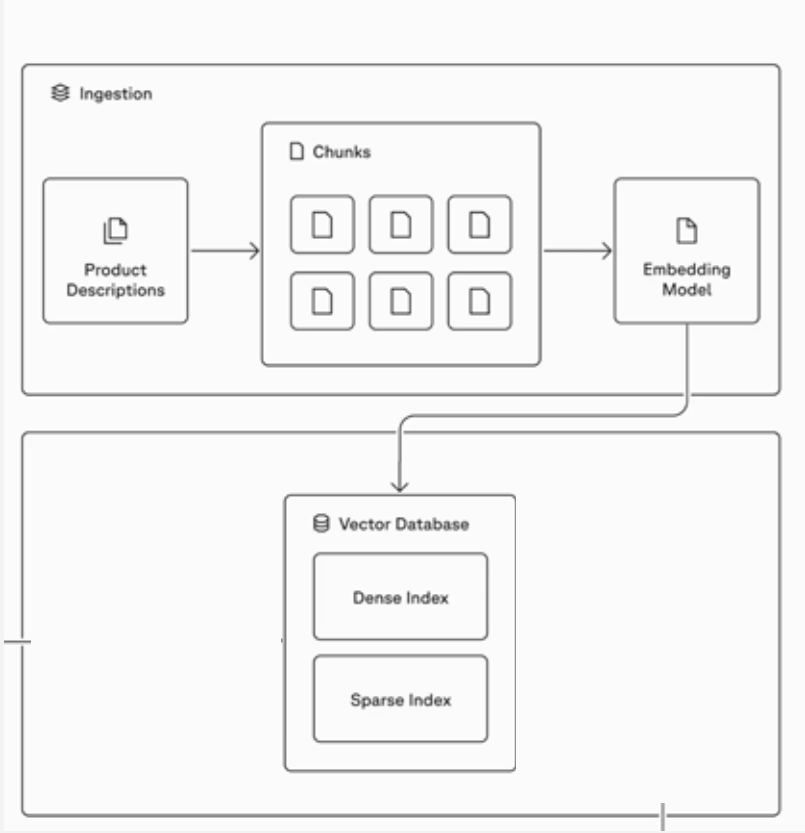
```
king − man ≈ queen − woman
```

Contextual Embeddings

- Vector depends on the sentence
- Meaning changes with context

`river bank ≠ financial bank`

Preparing Data For Retrieval



Chunking

table

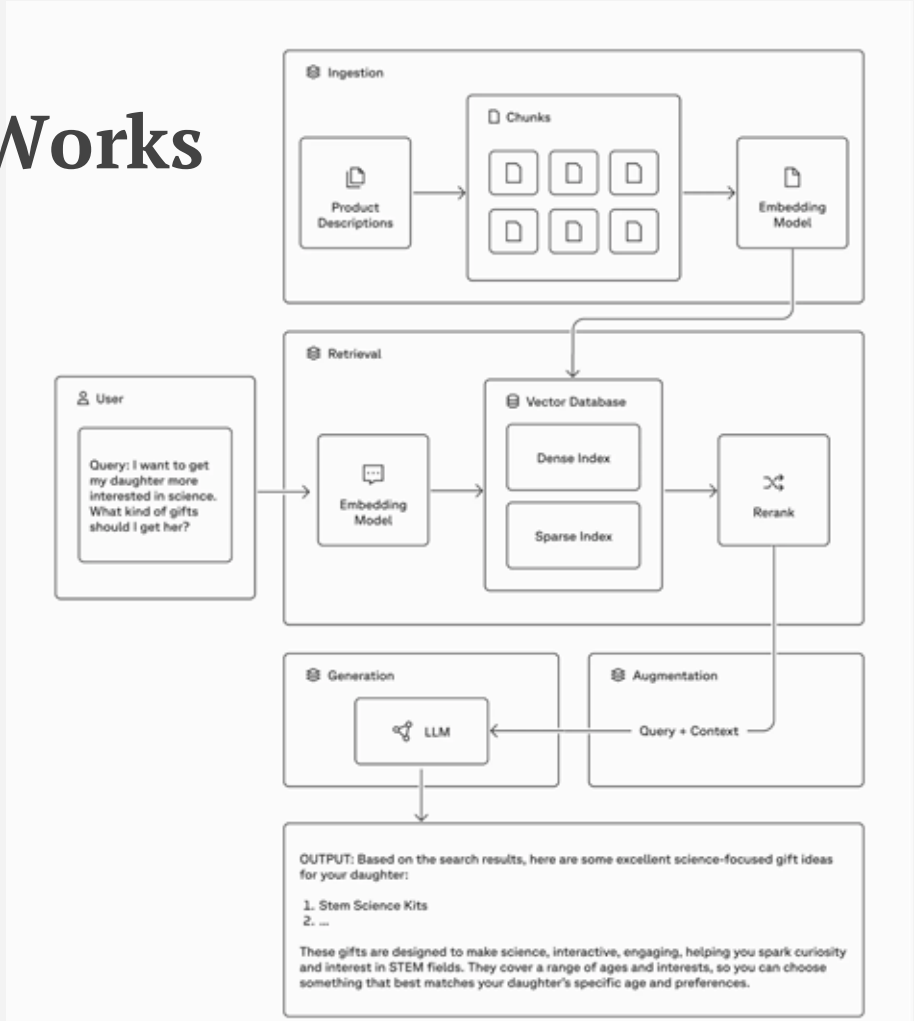
lecture

dialogue

documentation

RAG Demo

How RAG Works



AI Agents

The Principles

Spectrum of AI Agency

Most of the times AI Workflow is good enough!

Feature	AI Workflow	AI Agent
Goal & Scope	Operates within strict boundaries to complete a specific, pre-defined task (e.g., "summarize").	Works toward a general objective (e.g., "improve this") and figures out the necessary steps itself.
Workflow	Follows a flow based on the tools and instructions given to it.	Constructs its own workflow, planning and changing strategies dynamically as it progresses.
Decision Making	Decides **how** to execute the specific instruction given.	Decides **what** actually needs to be done, including defining its own sub-tasks and priorities.
Tools	Uses a closed, fixed set of pre-provided tools.	Selects tools independently and can even write code to create new tools if necessary.
Role Metaphor	**"Skilled Executor"** - Like an employee or soldier waiting for specific orders.	**"Independent Manager"** - Like a leader who initiates, plans, and manages the process.

AI Agents

Functional

Code/Application generation

Cognition · Lovable · .txt · nustom · IGENT.AI · a·gem·o · Sutro · OriginAI · marblism · poolside · Magic · Pythagora

Customer Support / Success

Gradient Labs · neople · zowie · M1 · parloa · Freeday · fini · PolyAI

Quality assurance

QA.tech · octomind · Myia · LogicStar · MAIHEM

GTM

11x · airs · Momentum · Vendi · salesforge · Synthflow · SIERRA · luna.ai

Security

Qevlar AI · AIKO · Tracecat · aikido · splx

Vertical

Legal

Leya · Lawhive · LegalFly · Wordsmith · Qura AI · Avantia

Finance

FINSTER AI · DECKMATCH · ClaimSorted · Vivox AI · thema

Healthcare

Co:Helm · FIINN · phare health · TORTUS · Tandem

General

General Agents

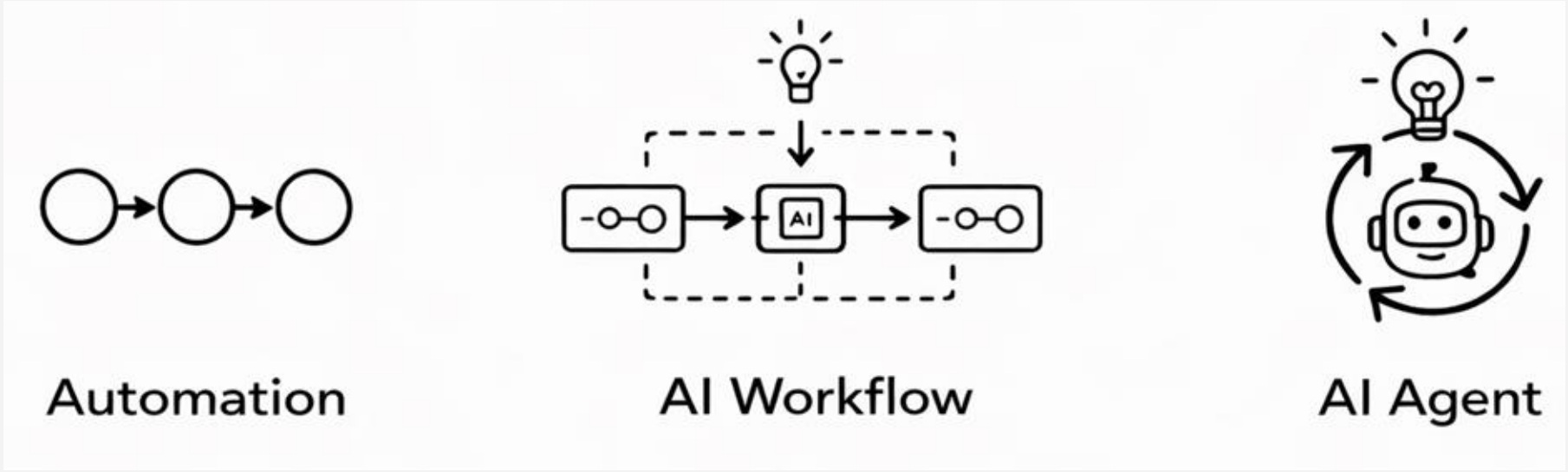
ZYLON · beam · AgentSea · AgentGPT · H · askui · Kili · LEMNI · occam.ai

Dawn PortCos building in AI

data iku · FLOWX.AI · swim

dawn

Level of Automation - Engineering

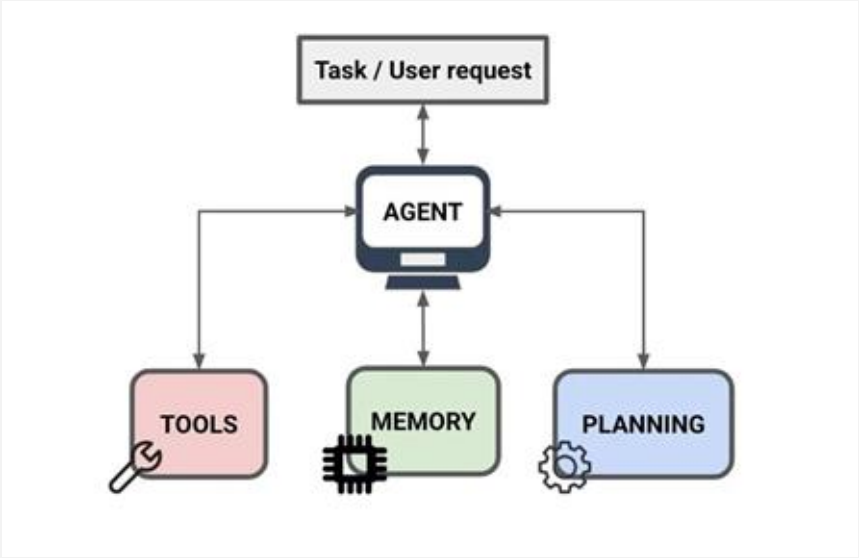


AI Agent Definition

An AI agent is a system that can perceive its **environment**, **reason** about it, and take **actions** to achieve specific **goals**.

Agent Components

1. LLM
2. Tools
3. Memory
4. Planning
5. User Request



Making Agents is HARD

So - Why Should we use it at all???

Some the reasons:

The "Control" Problem (Non-Determinism)

- **Probabilistic vs. Deterministic**
- **Cascading Failures:** Small hallucinations in Step 1 become disasters by Step 5.
- **Unpredictable**

Latency & User Experience

- **The "Thinking" Tax**
- **Patience is Low:** Users expect "chat speed" (<2s), not "agent speed."
- **Infinite Loops:** "Retry" logic can spiral, burning thousands of tokens in minutes.

Safety & Security

We will elaborate more during this course

making them dang

Context & Memory

- **Polluted History:** A mistake in Turn 3 becomes "truth" for Turn 4
- **The Fix:** Requires sophisticated state management, not just raw chat logs

AI Agents Prompt Engineering

ReAct = Reasoning + Acting

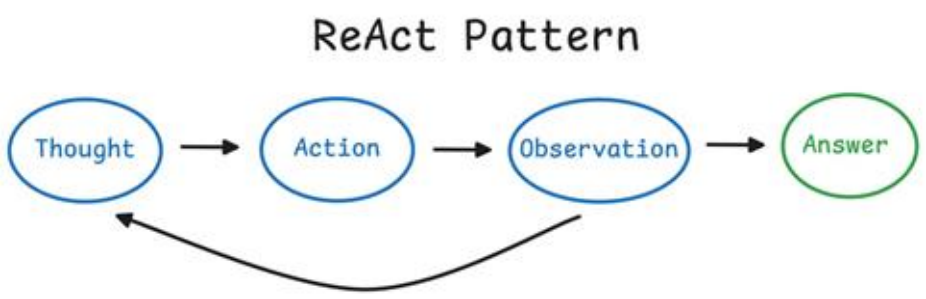
Prompt engineering technique that enables Large Language Models (LLMs) to solve complex tasks by combining **verbal reasoning** with **executable actions**

It was introduced in the paper *ReAct: Synergizing Reasoning and Acting in Language Models* (Yao et al., ICLR 2023).

ReAct Core Mechanism

ReAct forces the model to follow a specific three-step loop:

1. **Thought:** The model reasons about the current state and decides what information is missing.
2. **Action:** The model emits a specific command (e.g., `Search[query]`, `Calculator[expression]`) to retrieve external information.
3. **Observation:** The environment (or a parser) executes the action and returns the result to the model context.



* This loop repeats until the model has enough information to provide a **Final Answer**.

ReAct Demo

Responsible AI

Understanding LLM Limitations
(VERY PARTIAL LIST)

1. LLMs are Pattern Matching Machines

LLMs predict the next word based on patterns in their training data - like "autocomplete on steroids"

They are **"Plausibility Engines":** The output is optimized for what sounds plausible, not what is true or correct

This means:

- ➤ They excel at producing fluent, convincing-sounding text
- ➤ They can sound confident while being completely wrong
- ➤ Plausible ≠ True

2. The Capabilities Landscape

LLM capabilities are uneven: Impressive performance on hard tasks doesn't guarantee success on seemingly easier ones

Can do:
Pass the bar exam
Write sophisticated legal briefs
Analyze complex documents

Struggle with:
Verifying if its citations exist
Basic arithmetic in context
Simple spatial reasoning

Implication: You cannot assume competence transfers. Each task type requires verification

3. Hallucinations

LLMs generate **plausible-sounding** that might be false information

Air Canada Chatbot (2024)

Chatbot invented a bereavement policy

- Told customer: refund available after travel
- Real policy: must apply before booking
- Company held liable for chatbot's error

Deloitte Report (2025)

$290K government report contained AI errors

- Fake sources
- Fabricated court judgment quote
- Had to issue refund, revise report

Key point: Plausible ≠ True. Always verify AI-generated facts, citations, and policies

4. Privacy Risks

Your privacy expectations may not align with how **third-party LLM providers** operate.

INPUT Risk: Your prompts are discoverable
Sam Altman, OpenAI CEO (July 2025):
"So if you go talk to ChatGPT about your most sensitive stuff and then there's like a lawsuit or whatever, we could be required to produce that"
Risk: Prompts - even deleted ones - may be stored on provider servers and subject to subpoena

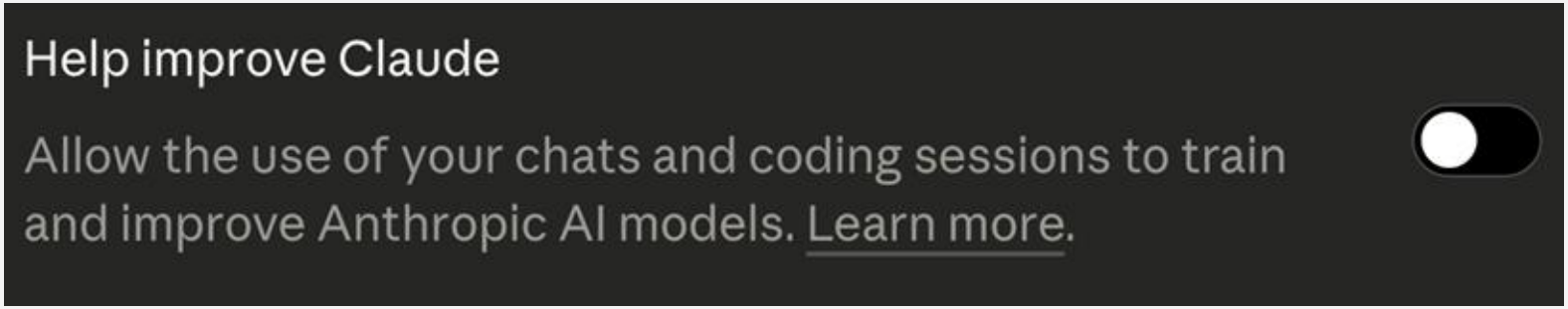
OUTPUT Risk: NYT Lawsuit (2023)
ChatGPT memorized NYT articles and could reproduce them verbatim

- ➤ 100+ examples of word-for-word copying
- ➤ Bypasses paywall entirely

Risk: Training data can be extracted.

Warning: What goes IN may be stored. What was trained ON may come OUT

4. Privacy Risks - con’t



Responsible Use: Some Core Issues

Pattern Matching Machines

Hallucinations

Capabilities Landscape

Privacy Risks