

# Using Generative AI in Research and Teaching: A Hands-on Workshop

Rafael  
Vazquez

Space Surveillance  
Chair



December 2024  
Politecnico di Milano



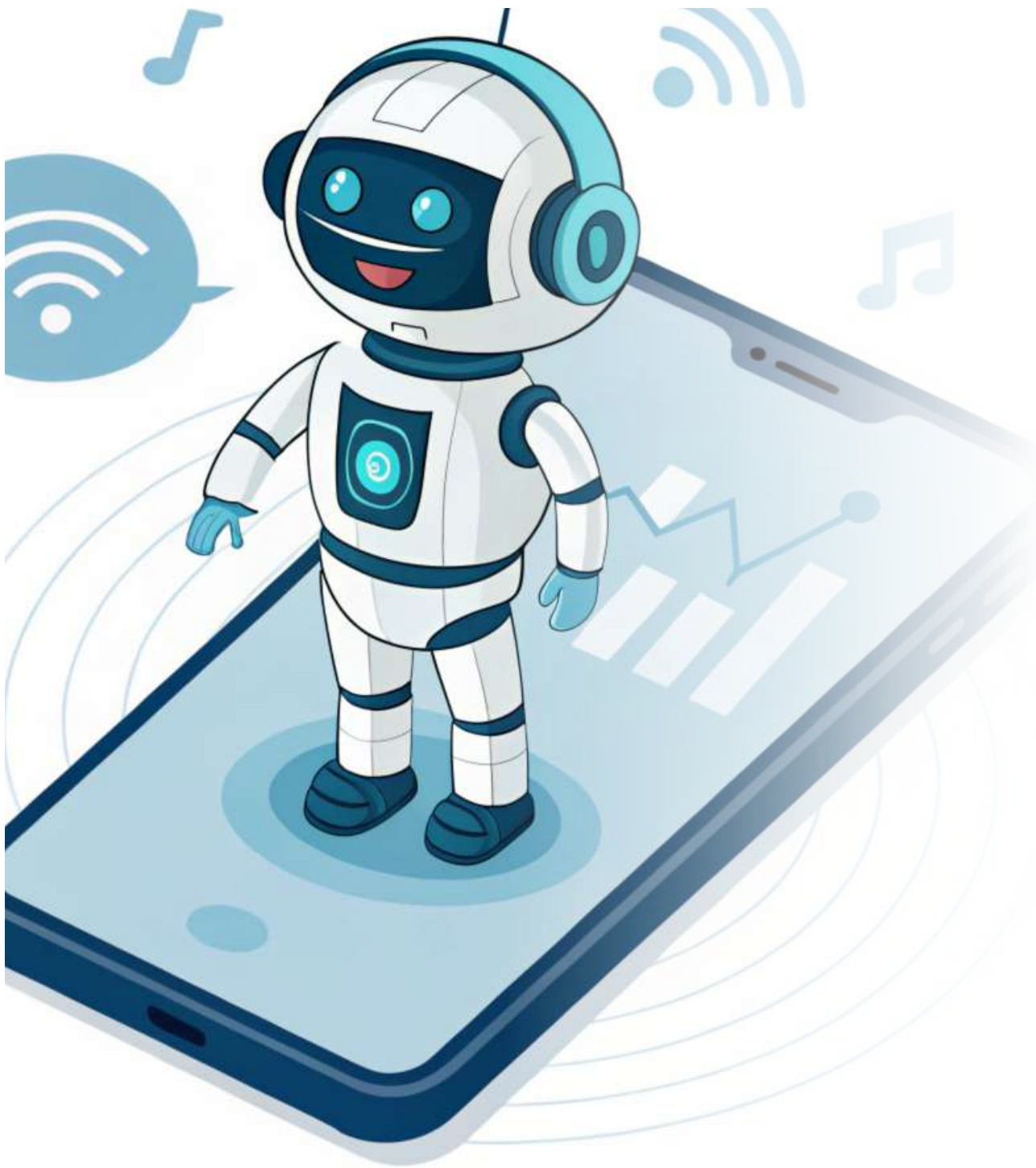


## Age of AI

- From Typing to Talking: The Interface Becomes “Human”
- From Describing to Understanding: AI “Comprehends” What It Sees
- From Spectators to Creators: AI Transforms Us Into Content
- From Dreams to Reality: AI Eliminates Creative Barriers

# The Great Democratizer





---

From Typing to  
Talking: The  
Interface  
Becomes  
“Human”

---

# Example

## ChatGPT Advanced Voice



**1. Engineering English Pronunciation** "I'll say some common engineering terms, and you'll tell me if I pronounced them correctly and help me with the right pronunciation. One by one: sensor, thermal, voltage, calibrate, frequency."

**2. Italian-Style Microfiction with Voices** "Create a short story in Italian about Professor Bianchi, an aerospace engineering professor at the Politecnico di Milano, who is drowning in administrative paperwork. Write it in the style of Italo Calvino or Umberto Eco, with two distinct voices: the tired, exasperated professor (speaking in a deep, weary Milanese accent) and the cheerful, efficient AI assistant (with a light, melodious voice). The story should show how AI helps him rediscover his passion for teaching."





From Describing to  
Understanding: AI  
“Comprehends”  
What It Sees





Prompt to Gemini 1206 (**no filters, T=1**): Write a humorous and critical commentary on the image of the Starlink satellite deployment. Focus on the potential negative consequences of filling low Earth orbit with thousands of satellites. Channel your inner satirist and don't be afraid to roast Elon Musk and his ambitions in the process. Make it entertaining

<https://youtu.be/a0IPFs9YHX8>





# From Spectators to Creators: AI Transforms Us Into Content



# Example

<https://youtu.be/tBxDzBlqiqo>

**11ElevenLabs**

**11eyGen**

---

# From Dreams to Reality: AI Eliminates Creative Barriers

---



# Example

<https://youtu.be/-vldYpeqLLk>

 **Claude**  
3.5 Sonnet

 **Suno**





**First, they replaced the writers, and I kept silent because I did not write for a living.**

**Then, they replaced the programmers, and I did not speak because code did not run through my veins.**

**Then, they replaced the artists, and I said nothing because my hands did not create art.**

**Then, they replaced the musicians, and I did not protest because music only lived in my heart, and I did not live from it.**

**And finally, they replaced us, the teachers, and by then the silence was deafening. No one remained to teach the new generations the value of the human voice, lost in the digital echo of AI.**

**There was no one left to remind us why we should speak, nor what to say.**

Respectfully inspired by the famous and terrible quote from Martin Niemöller, "First they came for the socialists..."



# Goals of this workshop

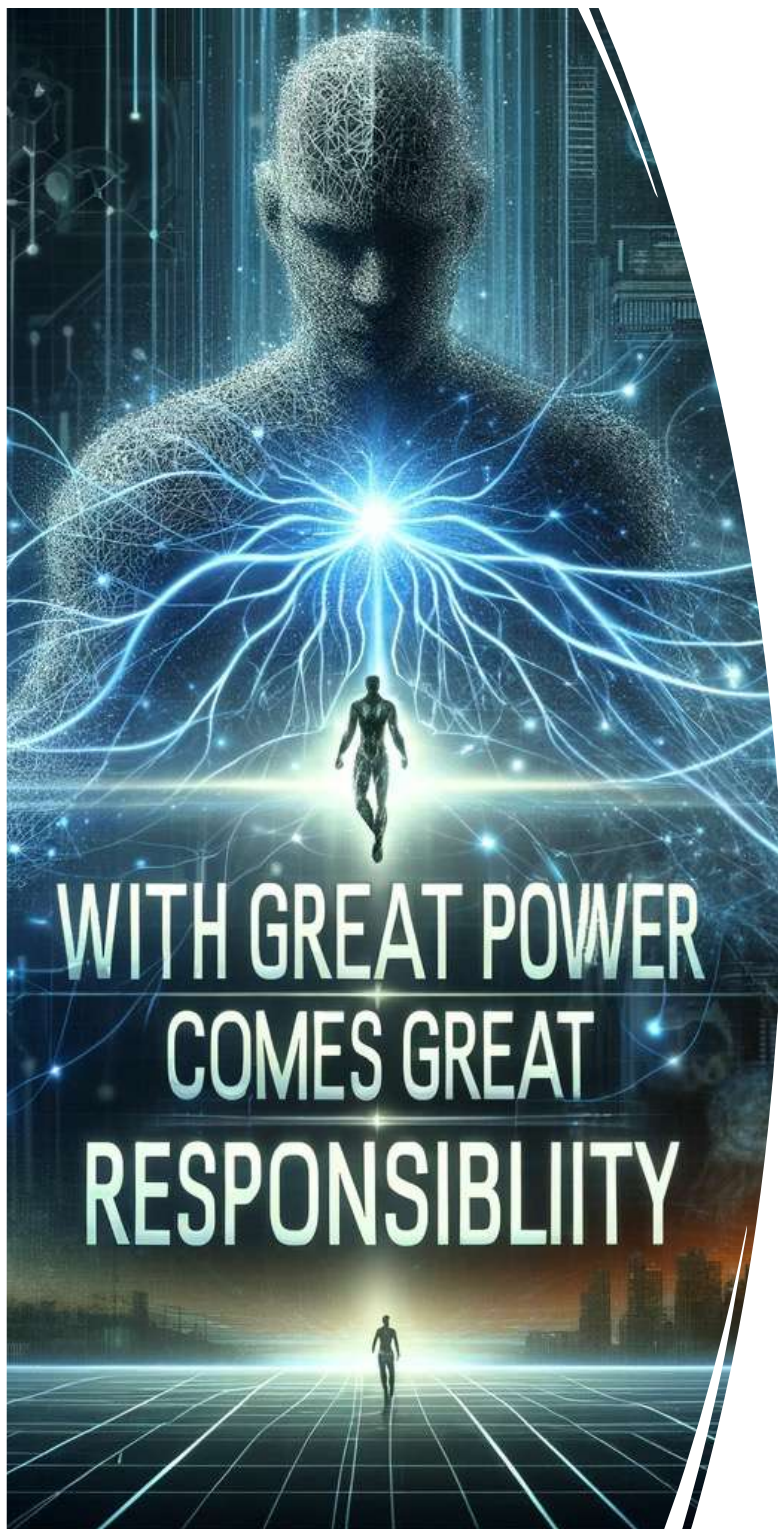
---

## *Accelerating Teaching, Research and Management*

- How LLMs work
- Prompt engineering with real practical cases of academic interest
- Knowledge of the current LLM ecosystem in 2024 and comparison of tools
- Limitations of LLMs
- Generating documents / presentations / code
- Other use cases in teaching and research environments

**(you may propose use cases for day 2)**





# Framework

---

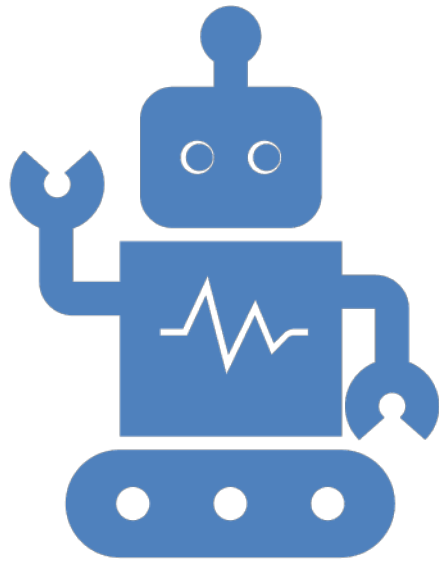
## *Responsible and Safe Use*

- Effective integration of AI in different tasks
- Balance between innovation and confidentiality
- Understanding of real capabilities and **limitations**

## *Practical Cases*

- Adaptable to individual needs
- Learning by *doing*

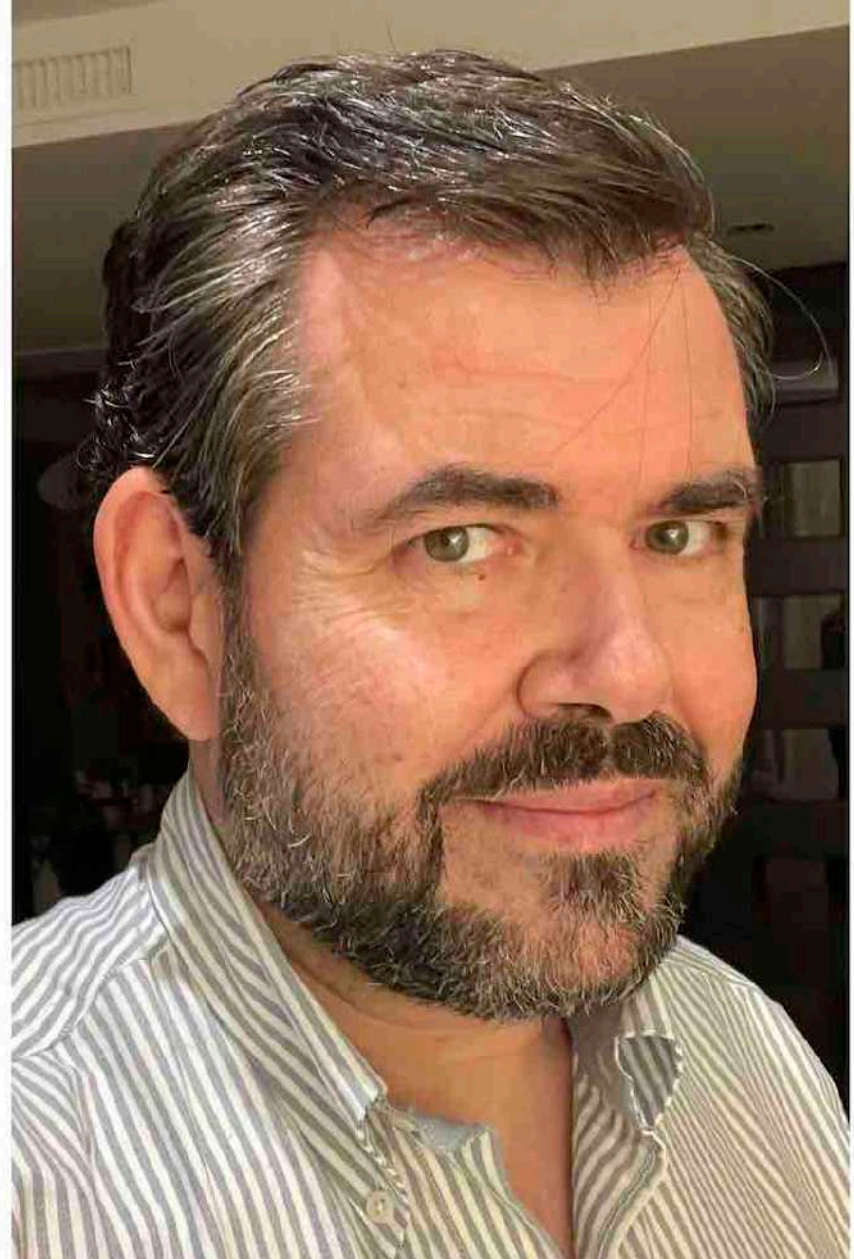




# Outline

1. **A personal journey into AI**
2. How LLMs work
3. Basic Prompt Engineering
4. Updated LLM Ecosystem (Dec 2024)
5. Limitations of LLMs
6. Advanced Prompt Engineering
7. Generating documents
8. Generating visual presentations
9. Generating code
10. Other use cases and tools
11. Closing and discussion

A personal  
journey into  
AI



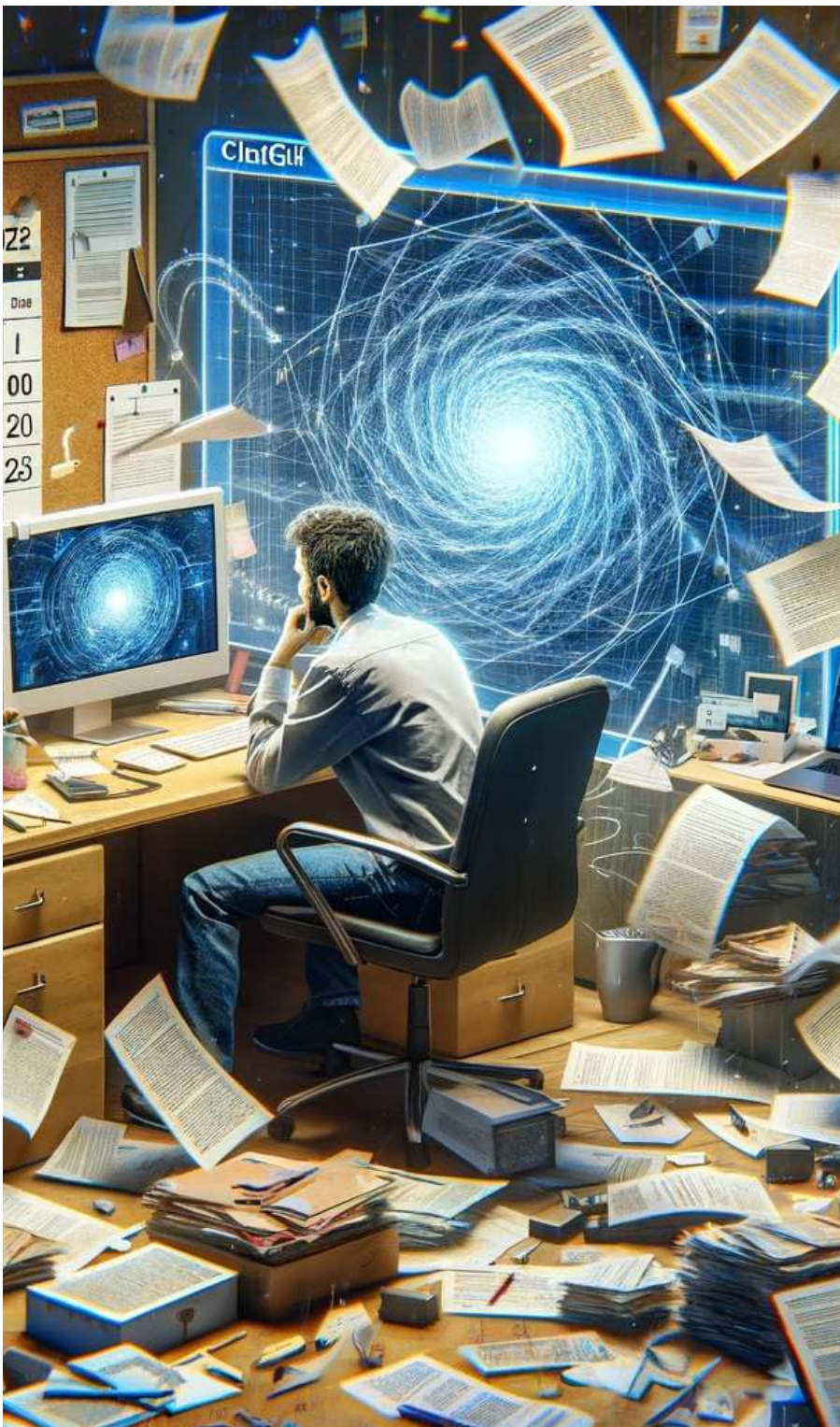


# Starting from healthy scepticism...

- Studied Neural Network in the 90s
- "Fear of Black Boxes"
- AI: another passing fad?







# First steps, first disappointments

- “ChatGPT” at the end of 2022
- First attempts: hallucinations, superficial writing
- Huge potential but it was not yet ready...

← ChatGPT thinks my office is like this



# The AI Tsunami

- Took a workshop just like this in June 2023
- AI = "*coming tsunami*"
  1. Ignore it at your own risk!

or

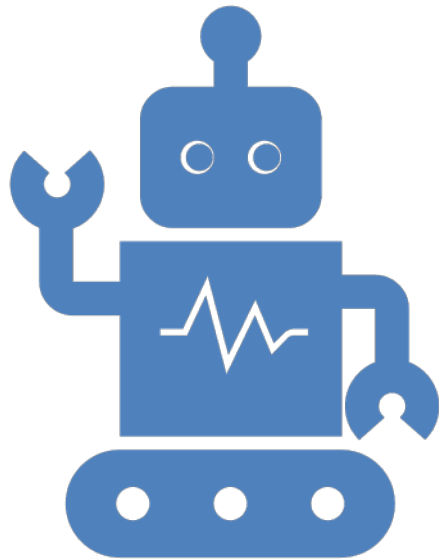
**2. Learn to swim!**





## From sceptic to faithful

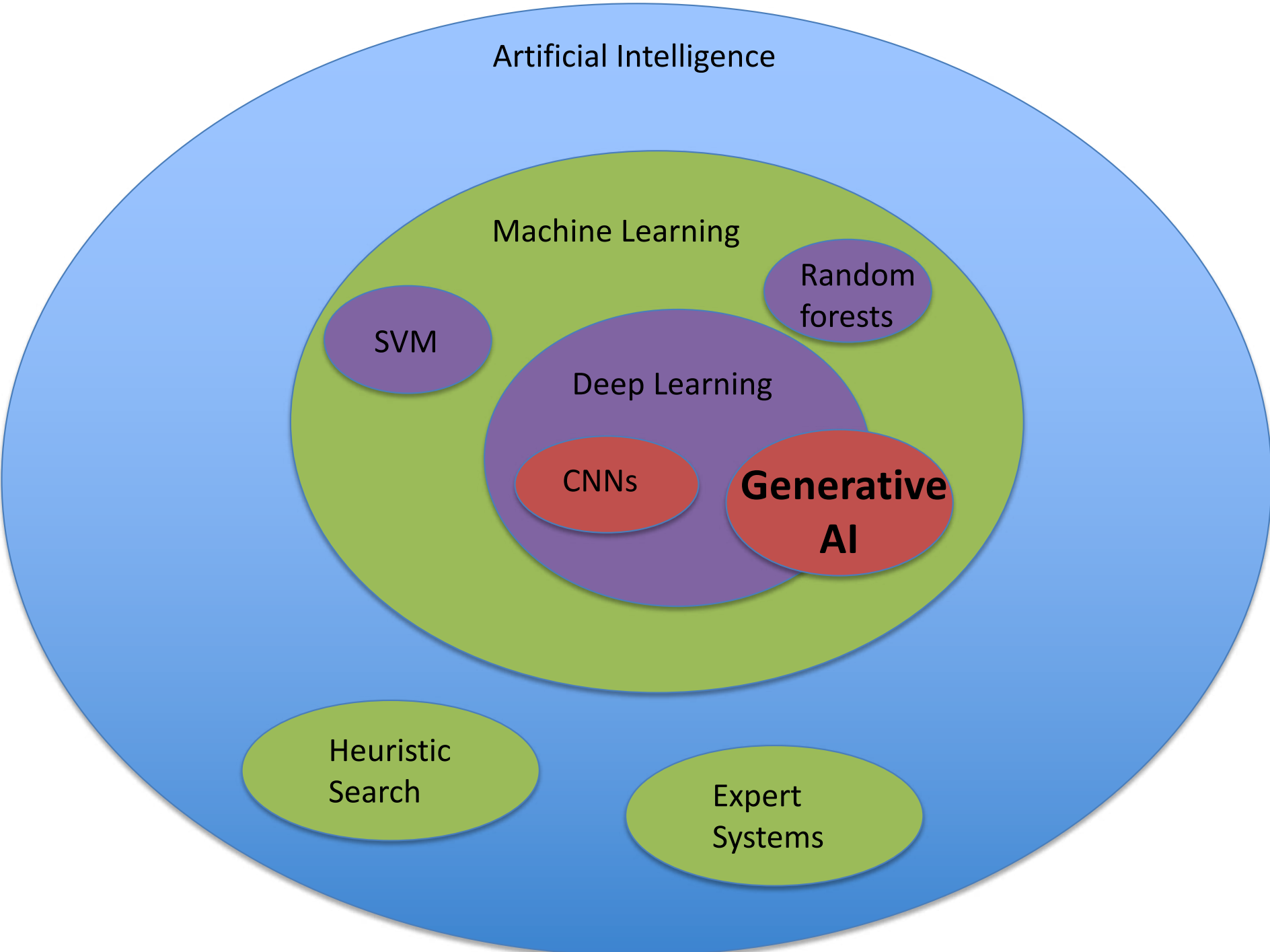
1. Adopted AI to develop new courses
2. Use of AI in routinary tasks (emails, recommendation letters, AE decisions...)
3. Use of AI in research writing (abstracts, conclusions, improve language...)
4. Improvement in technical tasks (e.g. programming)
5. Creating a presence in social networks (LinkedIn)
6. Creative use of AI
7. Teaching AI!



# Outline

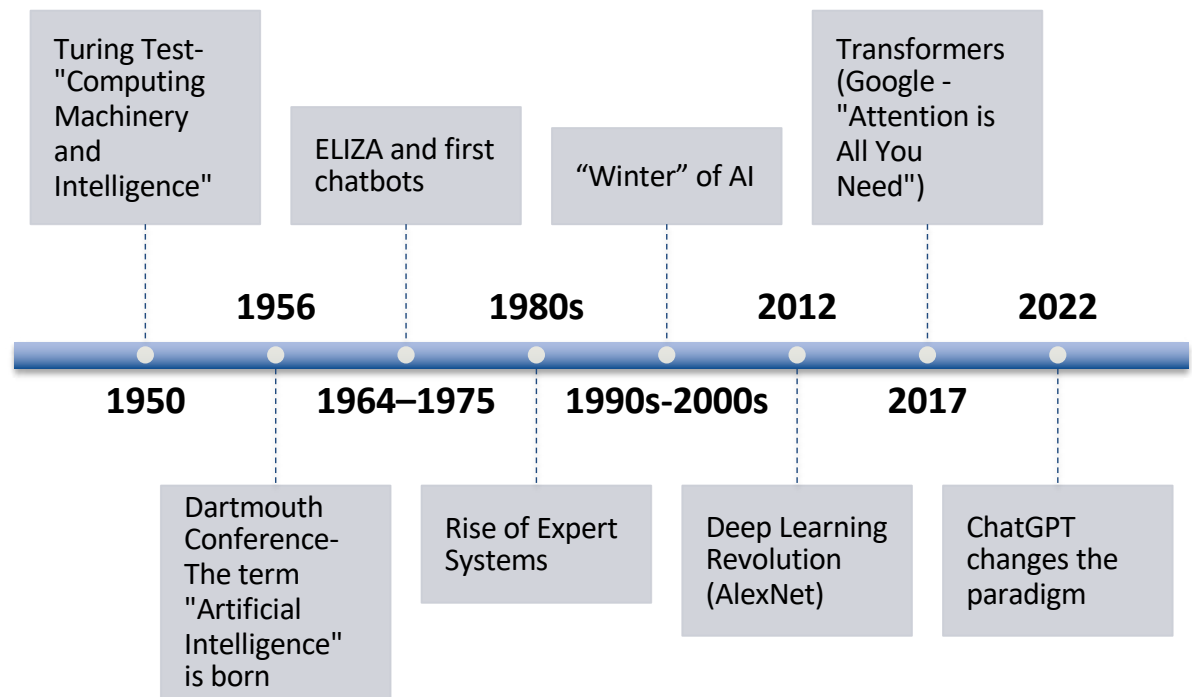
1. A personal journey into AI
2. **How LLMs work**
3. Basic Prompt Engineering
4. Updated LLM Ecosystem (Dec 2024)
5. Limitations of LLMs
6. Advanced Prompt Engineering
7. Generating documents
8. Generating visual presentations
9. Generating code
10. Other use cases and tools
11. Closing and discussion







# Historic Evolution of AI

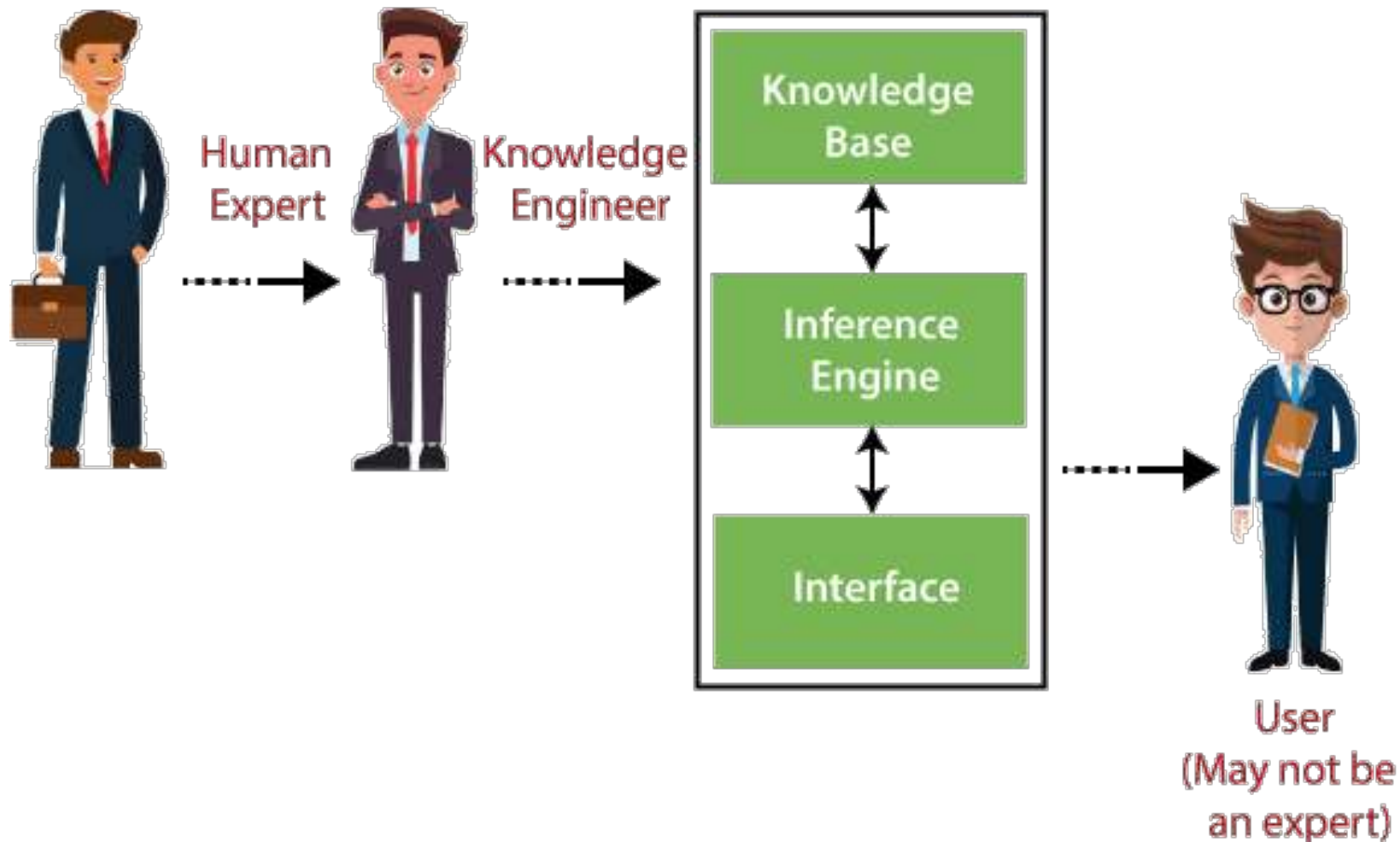




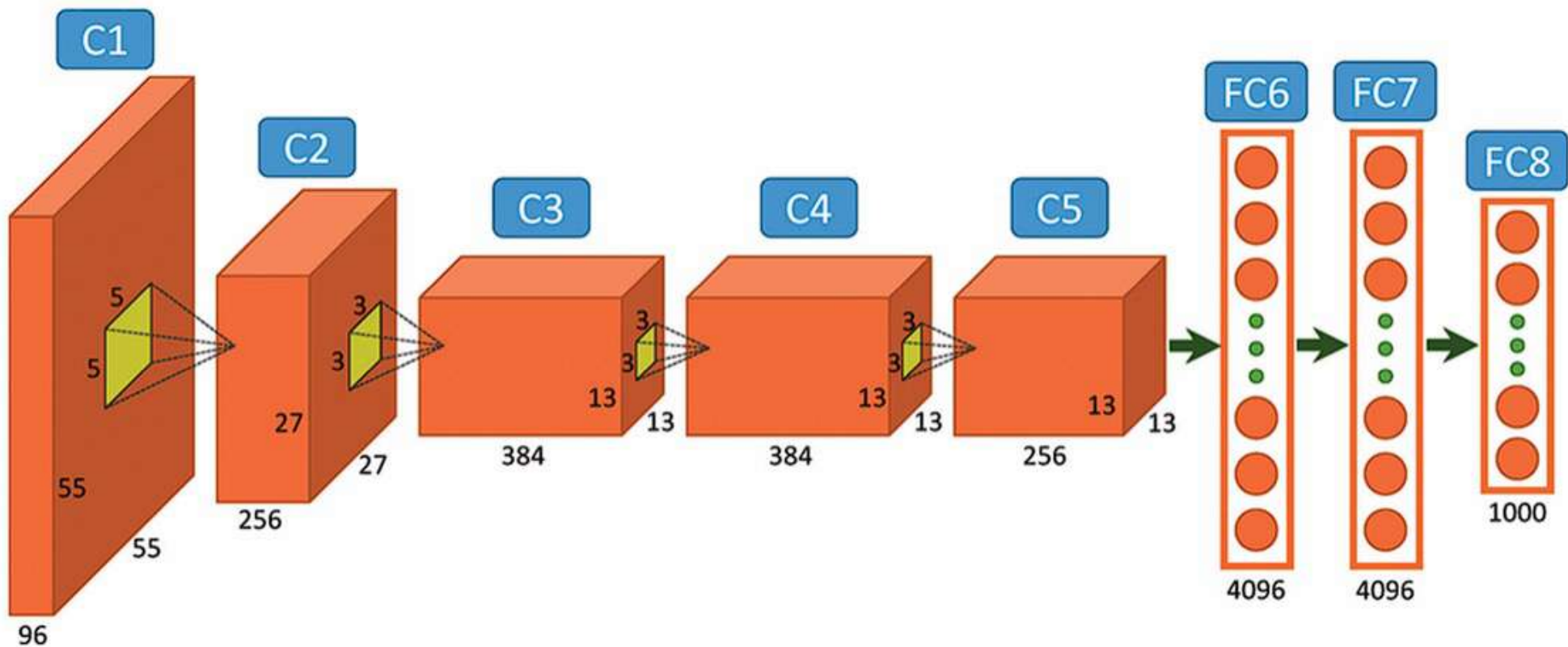


- **ELIZA (1964-1966)** ELIZA, created by Joseph Weizenbaum at MIT, was the first chatbot in history. It simulated being a psychotherapist through pattern matching and simple substitution. Its main technique was reformulating user statements as questions, creating a surprisingly effective illusion of understanding. The "**ELIZA Effect**" describes how people tend to anthropomorphize computers, attributing real understanding to them even while knowing their mechanical operation..

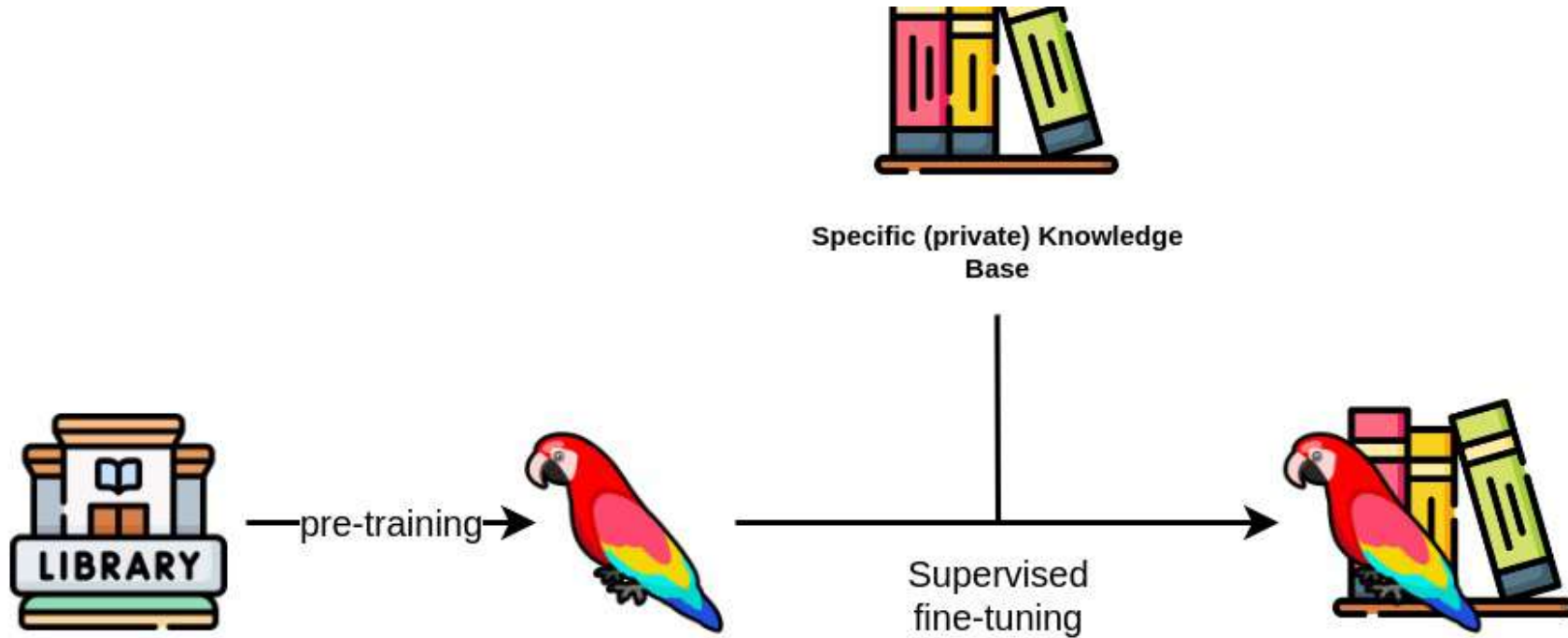
- **Expert Systems (1980s)** Expert systems attempted to codify human expert knowledge into explicit logical rules (if-then). MYCIN (medical diagnosis) and DENDRAL (chemistry) were prominent examples. Their fundamental limitation was "knowledge acquisition": the difficulty of converting expert knowledge into formal rules. This led to the "knowledge acquisition bottleneck," contributing to the first AI winter.







• **AlexNet (2012)** AlexNet, developed by Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton, marked the beginning of the modern deep learning revolution. Its victory in the ImageNet Challenge 2012 (reducing error from 26% to 15%) demonstrated the power of deep convolutional neural networks. Key innovations: use of ReLU, dropout and GPU for training. Its architecture of 8 layers and 60 million parameters seems modest today, but established the basic paradigm of current deep learning.

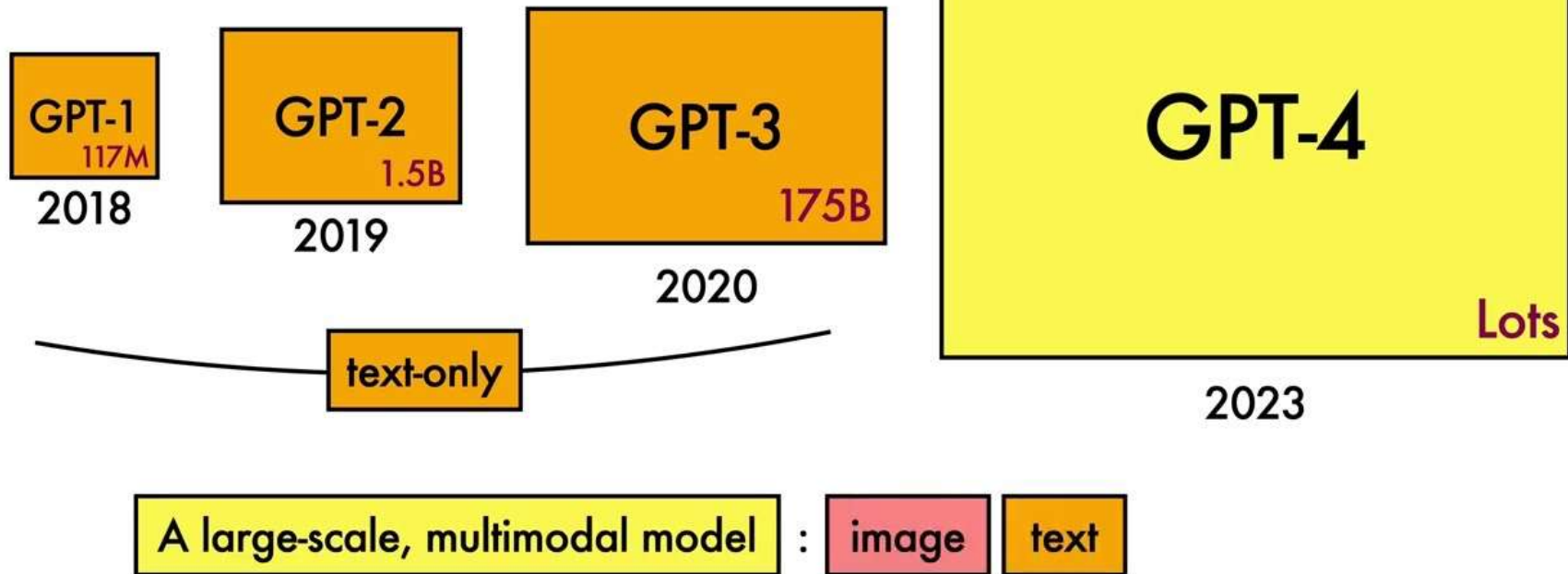


# Fundamentals of LLMs (Large Language Models)

## From Transformers to ChatGPT

- Transformer Architecture:
  - Revolution in NLP: *Attention mechanism*
  - Parallel processing
  - Contextual learning
- Training in three phases:
  - Pre-training with massive data
  - Fine-tuning
  - RLHF (Reinforcement Learning from Human Feedback)





ChatGPT

GPT= Generative Pre-trained Transformer

Tokens

94

Characters

382

We're no strangers to love  
You know the rules and so do I (do I)  
A full commitment's what I'm thinking of  
You wouldn't get this from any other guy  
I just wanna tell you how I'm feeling  
Gotta make you understand  
Never gonna give you up  
Never gonna let you down  
Never gonna run around and desert you  
Never gonna make you cry  
Never gonna say goodbye  
Never gonna tell a lie and hurt you

## Key concepts: Tokens

**How does an LLM "read" and "remember"?**

Tokens: The Basic Units

Not words, but text fragments

Examples:

- "hamburguesa" → ["ham", "bur", "guesa"]
- "artificial" → ["art", "ificial"]
- "intelligence" → ["intel", "ligence"]



# Tokens

GPT-4: ~128K tokens  $\approx$  96k words in English or 73k words in Spanish

<https://platform.openai.com/tokenizer>

The screenshot shows the OpenAI tokenizer interface. At the top, there are three tabs: "GPT-4o & GPT-4o mini" (selected), "GPT-3.5 & GPT-4", and "GPT-3 (Legacy)". The input text is "La lluvia en Sevilla es una pura Maravilla" and "The rain in Spain stays mainly in the plain". Below the input, there are two buttons: "Clear" and "Show example". The output shows the token count: 20 tokens and 86 characters. At the bottom, the text is displayed with individual tokens highlighted in different colors.

Tokens	Characters
20	86

La lluvia en Sevilla es una pura Maravilla  
The rain in Spain stays mainly in the plain

# Key concepts: "memory" of an LLM

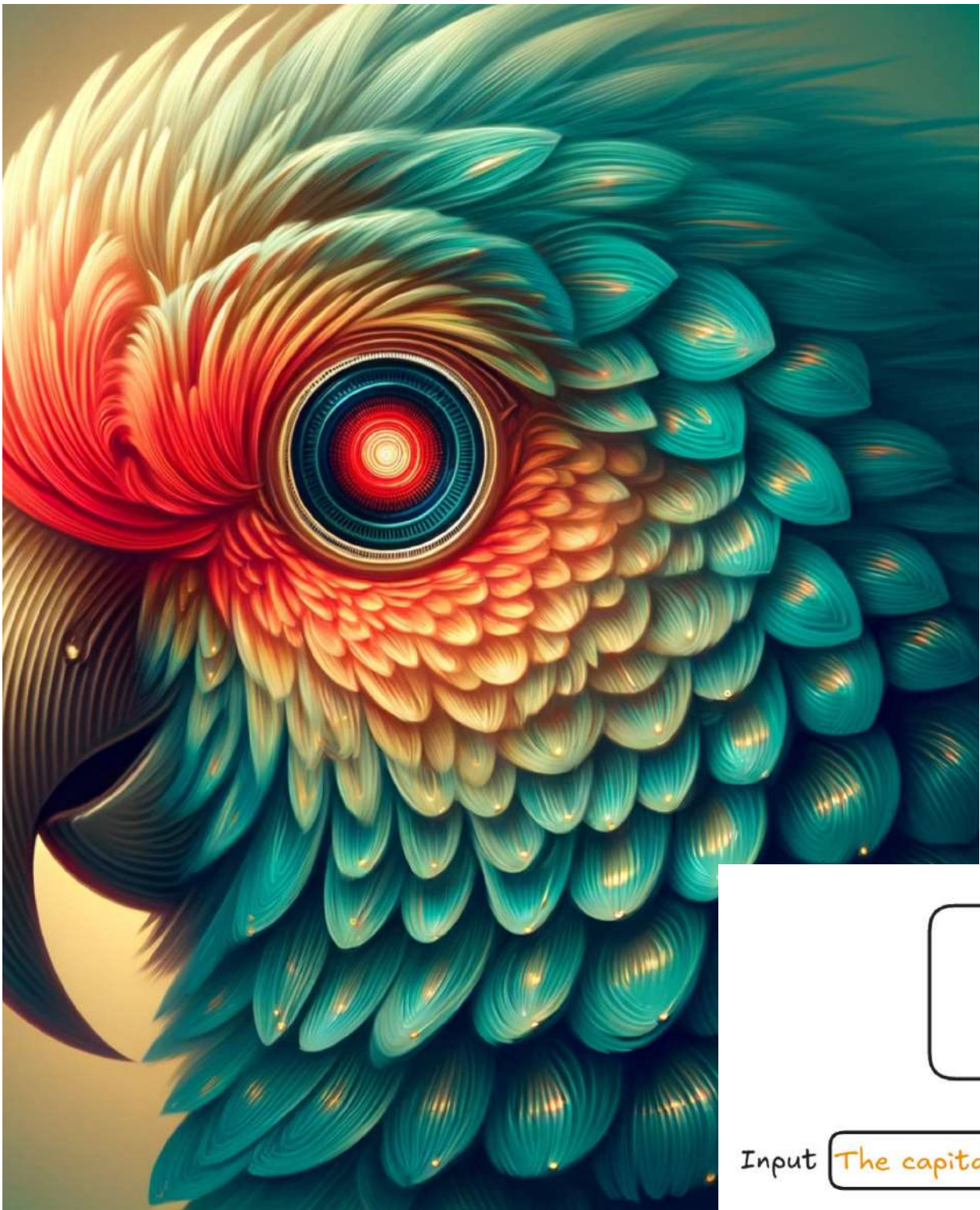
## The "Memory" System

- *Long-term Memory:*
  - Training data (base knowledge)
  - Does not change during conversation
  - Like a closed book that can only be read
- *Short-term Memory (Context Window):*
  - Like a temporary whiteboard
  - Only "remembers" the current conversation
  - Erased when starting a new conversation
  - DOES NOT TRAIN THE PRESENT MODEL! (it may help fine-tuning e.g. when choosing between two answers)
- *Limited by maximum number of tokens*

### *Analogy: A reference librarian*

- Long-Term Memory = All books in the library (can't add new ones)
- Context Window = Notes they can take while helping current patron

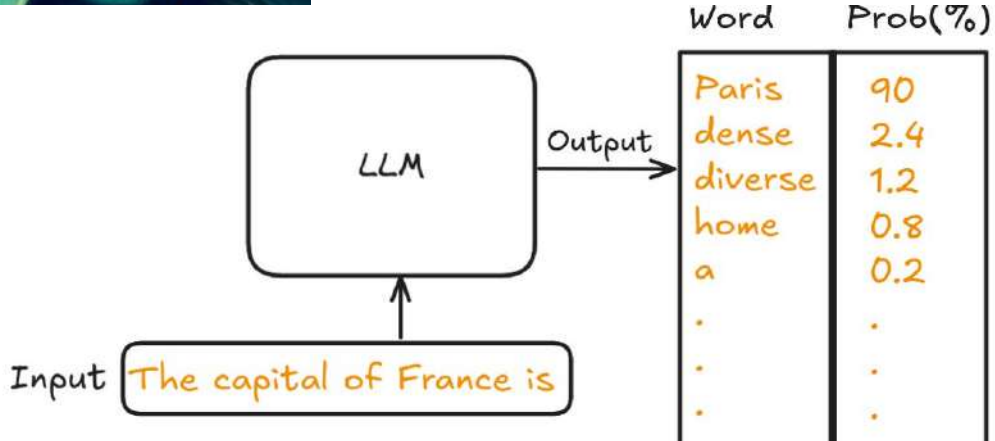




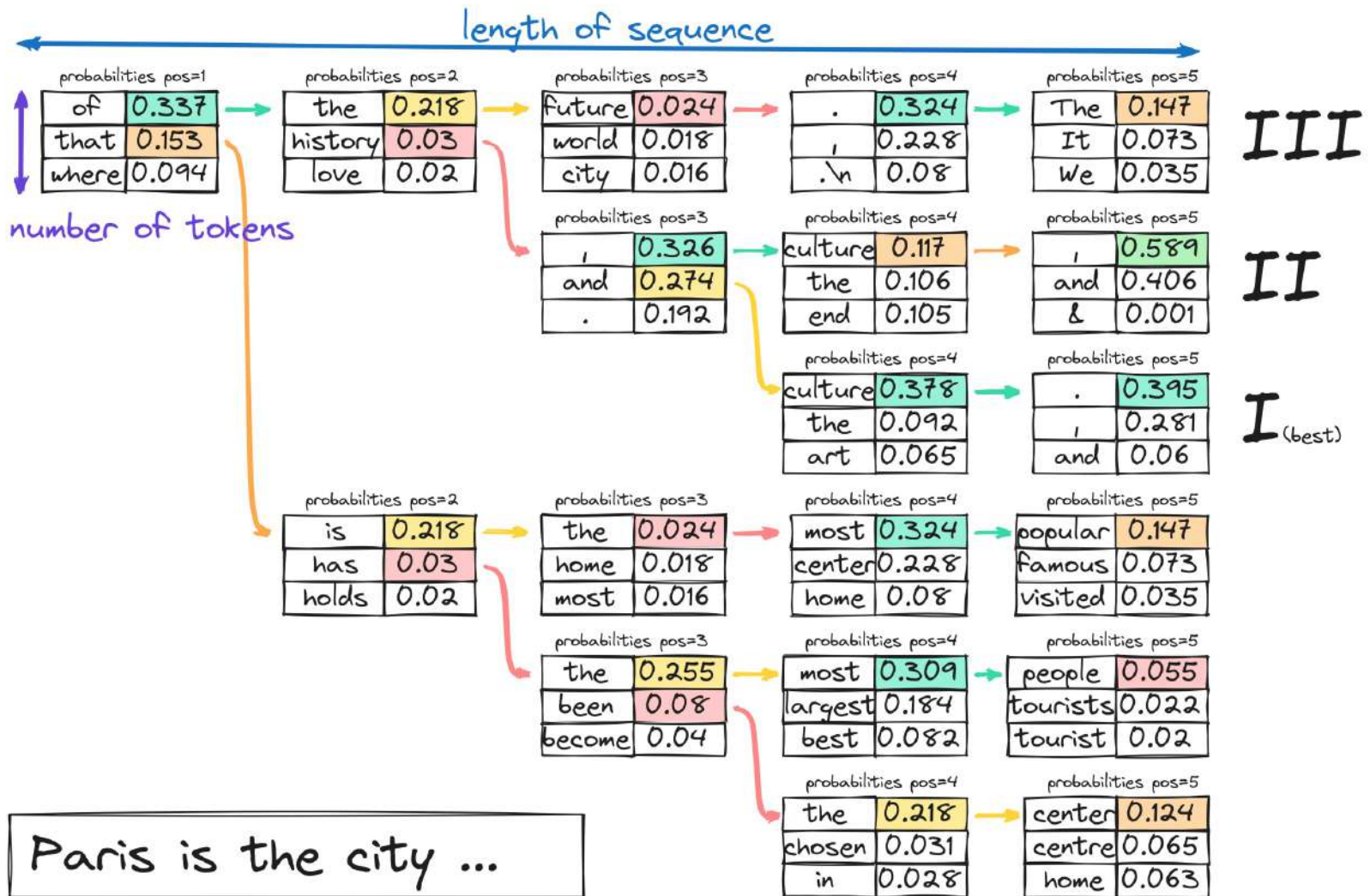
# How LLMs work

A "stochastic parrot"?

- Next token prediction based on probability
- Importance of context (context window)
- **Role of temperature:**
  - High = more creative but less accurate
  - Low = more consistent but less creative
- Not "understanding" but advanced statistical correlation



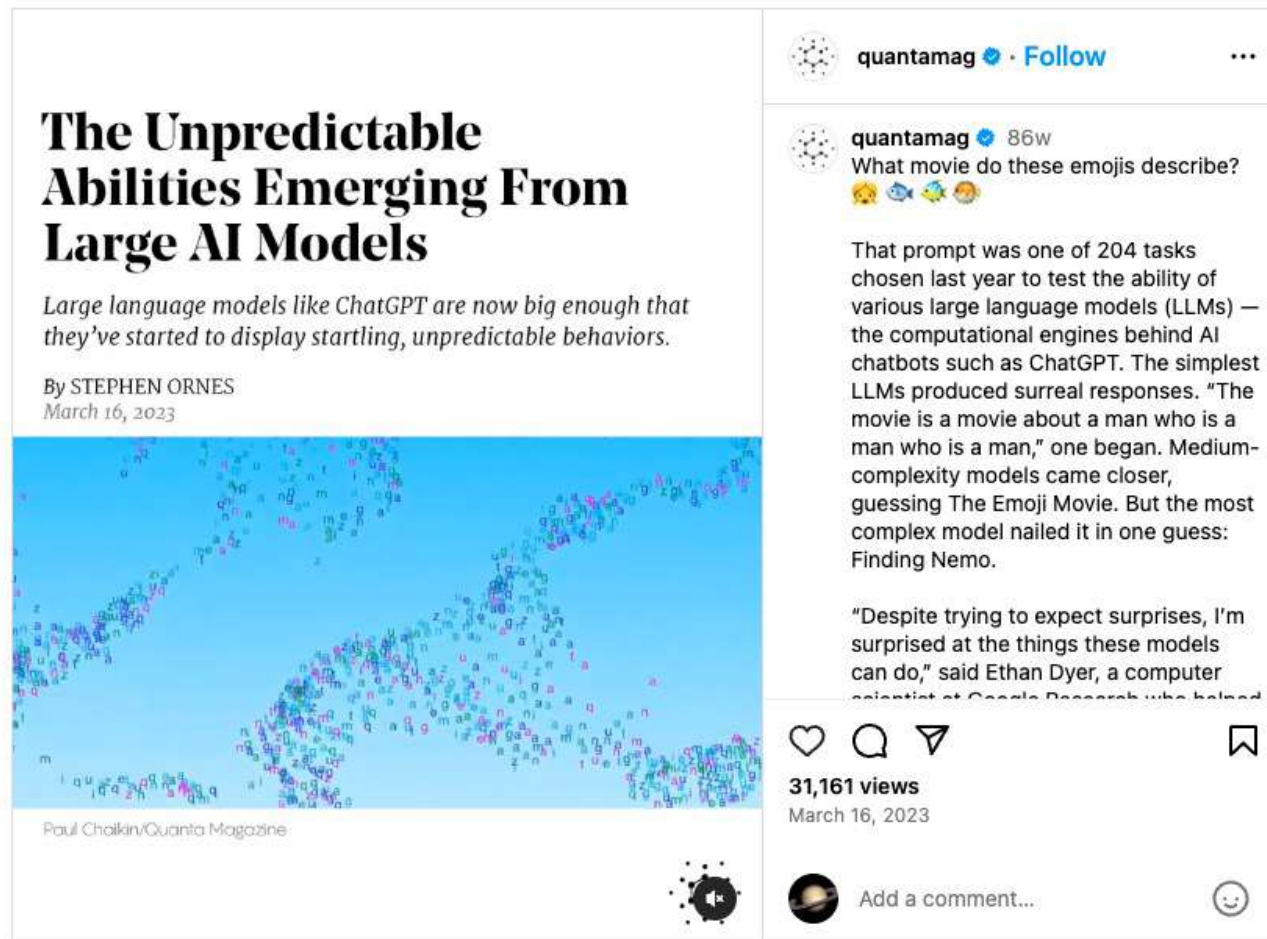
# How LLMs work





# Unexpected Responses

Token-by-Token prediction doesn't fully explain these capabilities



The screenshot shows a social media post from the account 'quantamag'. The post features a video player on the left and text on the right. The video player displays a title 'The Unpredictable Abilities Emerging From Large AI Models' by Stephen Ornes, dated March 16, 2023. The video content shows a large number of colorful characters (a, b, c, etc.) arranged in a pattern that resembles a map of the United States. Below the video, the text reads 'Paul Chaikin/Quanta Magazine'. The right side of the post contains the text of the video's content, which discusses a prompt 'What movie do these emojis describe?' with emojis of a yellow smiley face, a blue fish, and a yellow smiley face with a tongue sticking out. The text describes a challenge where 204 tasks were used to test various large language models (LLMs). It notes that the simplest LLMs produced surreal responses, such as 'The movie is a movie about a man who is a man who is a man, one began. Medium-complexity models came closer, guessing The Emoji Movie. But the most complex model nailed it in one guess: Finding Nemo.' It also includes a quote from Ethan Dyer, a computer scientist at Google Research, who said, 'Despite trying to expect surprises, I'm surprised at the things these models can do.'

**The Unpredictable Abilities Emerging From Large AI Models**

*Large language models like ChatGPT are now big enough that they've started to display startling, unpredictable behaviors.*

By STEPHEN ORNES  
March 16, 2023

Paul Chaikin/Quanta Magazine

quantamag · Follow

quantamag 86w  
What movie do these emojis describe?  
😊🐟😊

That prompt was one of 204 tasks chosen last year to test the ability of various large language models (LLMs) — the computational engines behind AI chatbots such as ChatGPT. The simplest LLMs produced surreal responses. "The movie is a movie about a man who is a man who is a man," one began. Medium-complexity models came closer, guessing The Emoji Movie. But the most complex model nailed it in one guess: Finding Nemo.

"Despite trying to expect surprises, I'm surprised at the things these models can do," said Ethan Dyer, a computer scientist at Google Research who helped

31,161 views  
March 16, 2023

Add a comment...



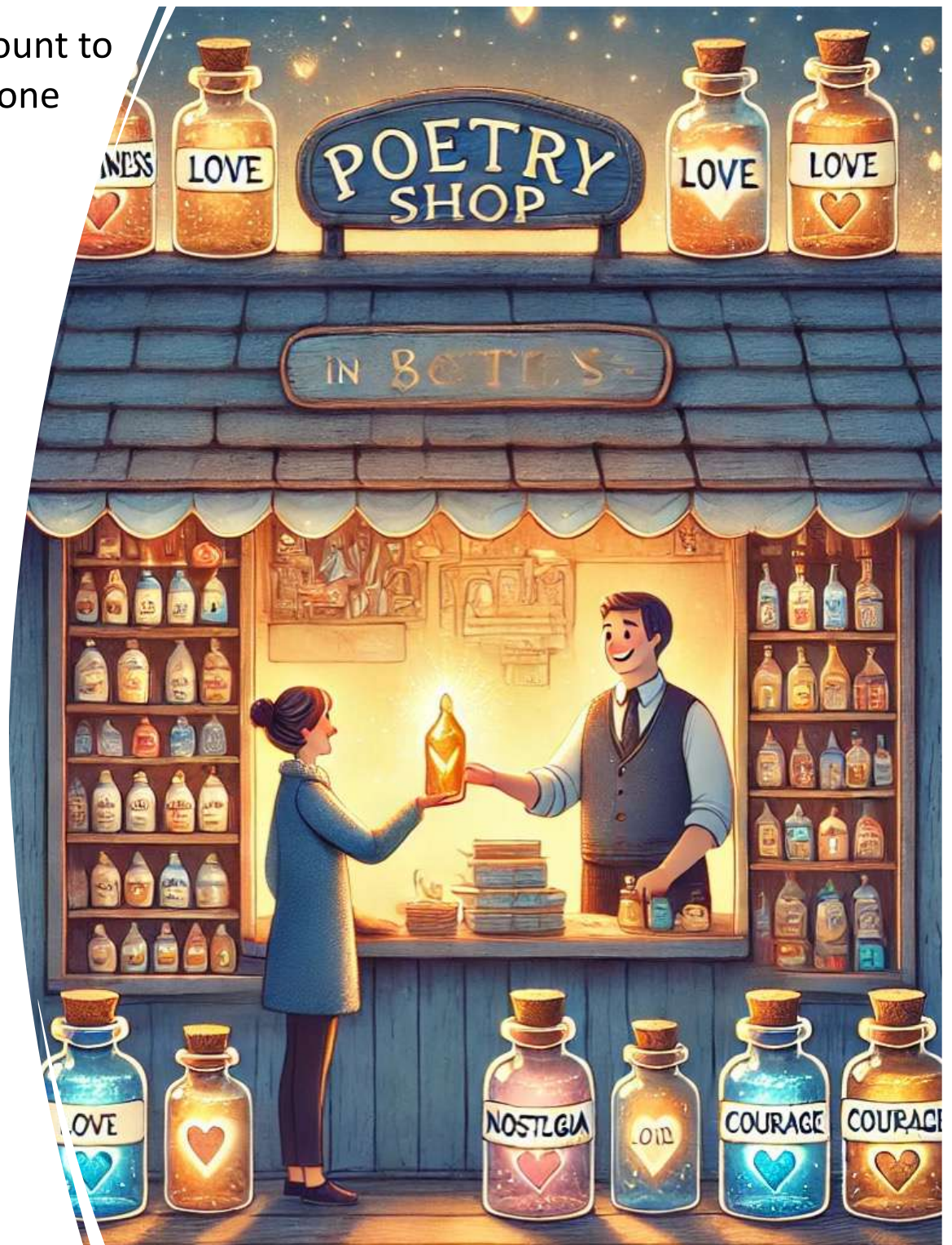
<https://chat.com> -> Use your Google account to create OpenAI account if you don't have one

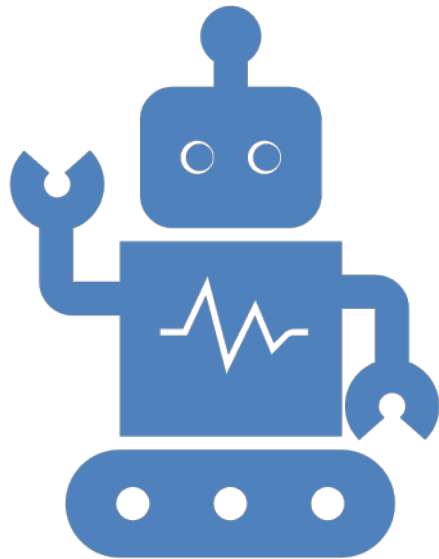
# First Exercise

## Exercise: The Poetry Shop

- Instructions for participants: You own a small poetry shop that sells emotions in bottles. Ask ChatGPT to help you write a description of your most popular product. Use exactly this prompt<sup>1</sup>:
- 'I own a small poetry shop that sells emotions in bottles. Please write a brief description (2-3 sentences) of our bestseller product.'
- Compare between participants!

<sup>1</sup>**Prompt:** The text you write to tell an AI what you want it to do





# Outline

1. A personal journey into AI
2. How LLMs work
3. **Basic Prompt Engineering**
4. Updated LLM Ecosystem (Dec 2024)
5. Limitations of LLMs
6. Advanced Prompt Engineering
7. Generating documents
8. Generating visual presentations
9. Generating code
10. Other use cases and tools
11. Closing and discussion

# Prompt Engineering: The *art* of communicating with AIs

*From basic instructions to  
professional results*





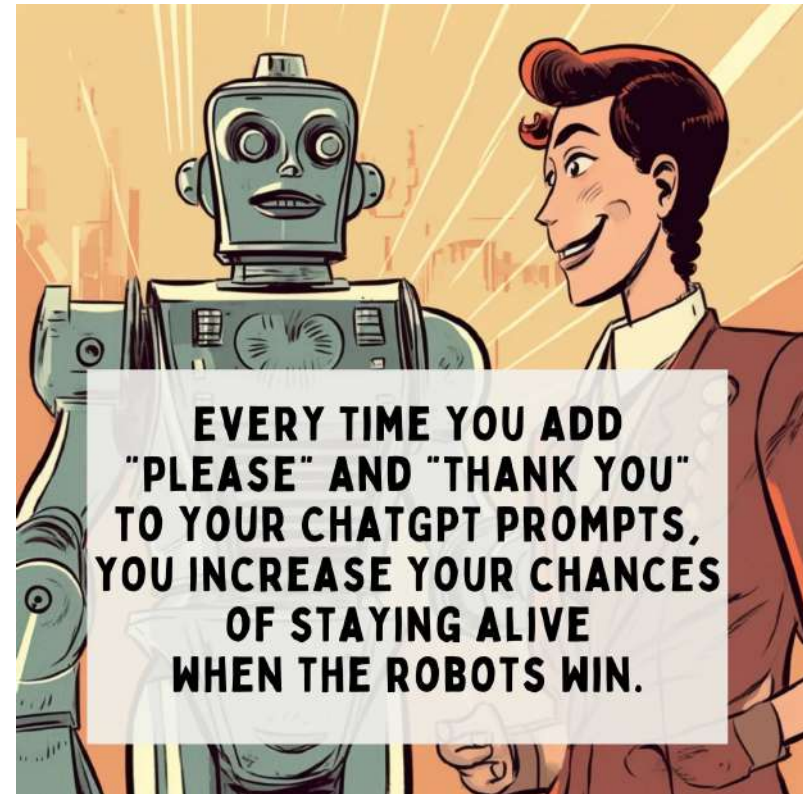
# What is a prompt?

- **The instruction we give to AI...** or more formally: A prompt is an instruction or question that guides the generation of responses.
- "Like giving instructions to a brilliant but... somewhat obtuse intern"
- Why does it matter? The quality of the output directly depends on the input!



# Anatomy of a good prompt

- Key components:
  - Specific role
  - Clear context
  - Specific instructions
  - Desired format
  - Constraints/limits
- Iteration/feedback until desired result is achieved
- Politeness?



## Specific Role

Here's an abstract of a paper about composite materials fatigue testing: [see next page, needs to be pasted]

Plus these authentic quick notes you jotted down while reading the manuscript: -

"interesting new approach to fatigue" -

"methodology needs more detail" - "great potential for aircraft maintenance" -

"graphs hard to understand" - "novel use of sensors" - "conclusions bit rushed" And

your informal thought: "Pretty good work overall, but needs to be presented more clearly for the aviation maintenance community"

Task: Using these real materials (abstract + notes + informal opinion), craft a structured formal review.



# Example continued

Abstract for example (AI made)

"Enhanced Fatigue Life Prediction in Carbon-Epoxy Composites Using Multi-Modal Sensor Arrays"

Carbon fiber reinforced polymers (CFRP) are increasingly critical in modern aircraft structures, yet their fatigue behavior remains challenging to predict accurately. This study presents a novel approach combining distributed fiber optic sensors and acoustic emission monitoring to detect precursors of fatigue damage in carbon-epoxy composites. We subjected 48 CFRP specimens to cyclic loading while simultaneously recording strain distributions and acoustic events. Our results demonstrate that the correlation between localized strain anomalies and specific acoustic signatures can predict fatigue failure 150-200 cycles earlier than conventional methods. Furthermore, we developed a machine learning model that achieved 87% accuracy in distinguishing between delamination and matrix cracking based on the multi-modal sensor data. These findings suggest a promising pathway for implementing real-time structural health monitoring in aerospace composites, potentially reducing maintenance downtime and improving safety margins.

## Clear Context

You're coordinating next month's departmental seminar series.

Objective: Craft an email announcing a guest lecture  
Context: NASA's Chief Engineer for the Artemis program has agreed to give a virtual guest lecture

Details: Dr. Sarah Chen, NASA's Artemis Chief Engineer, will give a 1-hour virtual lecture on 'Challenges in Lunar Landing Systems' on January 15th at 2 PM EST. You need to announce this to faculty and graduate students, highlighting the unique opportunity for Q&A with a leading industry expert.

## Specific Instructions

Objective: Transform some technical research into a compelling grant abstract

Instructions:

250-word grant abstract: - States the problem in first paragraph - Describes novel approach in second paragraph - Outlines expected impact in final paragraph - Uses language suitable for interdisciplinary reviewers - Includes one key metric or quantifiable outcome - References one relevant NASA/ESA strategic goal

[You need to attach paper in PDF or paste text of some research in space engineering, e.g.

<https://aero.us.es/rvazquez/papers/single-montilla.pdf> ]



## Desired Format

Objective: Create standardized documentation for lab equipment  
Required structure:

SECTIONS: - Equipment Overview (2-3 sentences) - Safety Precautions (bullet points) - Pre-Calibration Checklist (numbered steps) - Calibration Procedure (detailed steps) - Expected Values (table format) - Troubleshooting Guide (if/then format) - Sign-off Requirements (checklist)

Include PDF manual of lab equipment (for instance <https://www.vernier.com/files/manuals/acc-bta/acc-bta.pdf>)

## Constraints/Limits

Objective: Create a research highlight for department newsletter  
Constraints: - Maximum 150 words  
- Must include one technical diagram reference - No more than 2 technical terms requiring definition  
- Suitable for first-year engineering students - Include one real-world application - No mathematical equations

[use the same paper as before]

# Prompting Exercise

Objective: Transform a technical paper into a 15-minute conference presentation.

You are a keynote speaker at the International Astronautical Congress, need to create a presentation outline for your paper [attach previous paper].

Include:

-Role: Keynote speaker addressing mixed audience of academics and industry professionals

-Context: 15-minute slot followed by 5-minute Q&A

Instructions:

Convert key findings into 5-7 main points

Include 2 case studies

Add 1 industry application

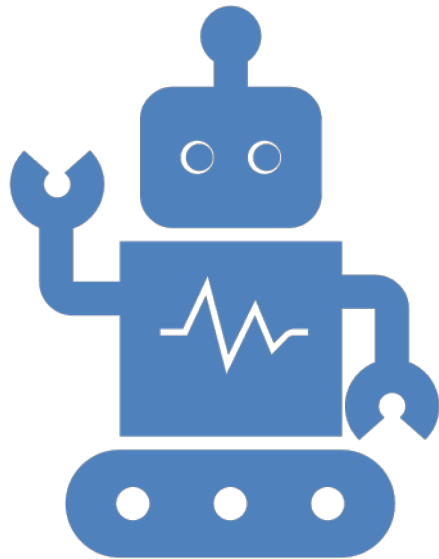
Format: Slide-by-slide outline with speaking notes

Constraints: - Max 12 slides

- No more than 2 equations per slide (use latex)

- Must include one visualization





# Outline

1. A personal journey into AI
2. How LLMs work
3. Basic Prompt Engineering
4. **Updated LLM Ecosystem (Dec 2024)**
5. Limitations of LLMs
6. Advanced Prompt Engineering
7. Generating documents
8. Generating visual presentations
9. Generating code
10. Other use cases and tools
11. Closing and discussion

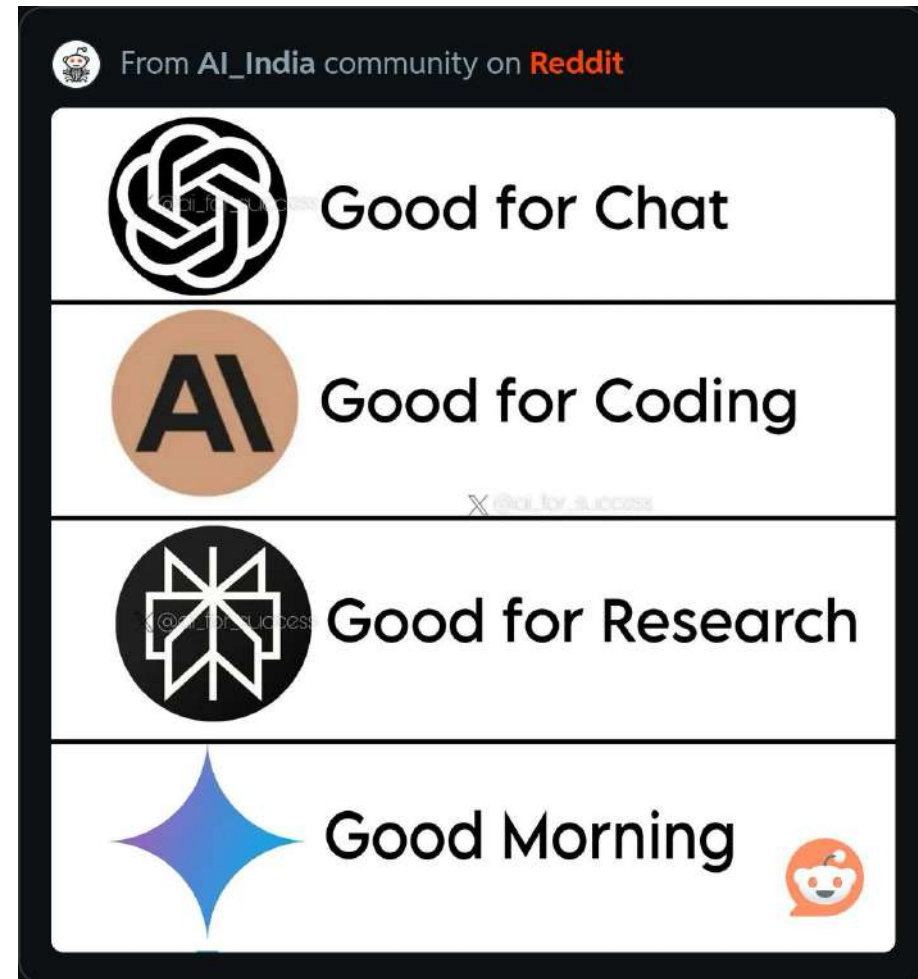
# Updated Ecosystem of LLMs (Dec 2024)

Free and paid tools available today:

- Comparison
- Prices
- Pros and cons of each tool

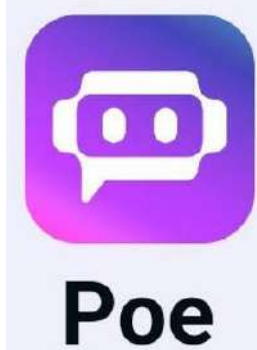
A partially subjective opinion

Test prompts in class!





OpenAI  
ChatGPT **4.0**



**Gemini**

 **Claude**  
3.5 Sonnet



**Copilot**

 perplexity



**Grok**



**LE CHAT  
MISTRAL**



# Ecosistema de LLMs

---

ChatGPT 4o, o1

---

Claude 3.5 Sonnet

---

Copilot

---

Mistral

---

Gemini 1.5 Pro (Exp 1206)

---

Perplexity

---

Poe

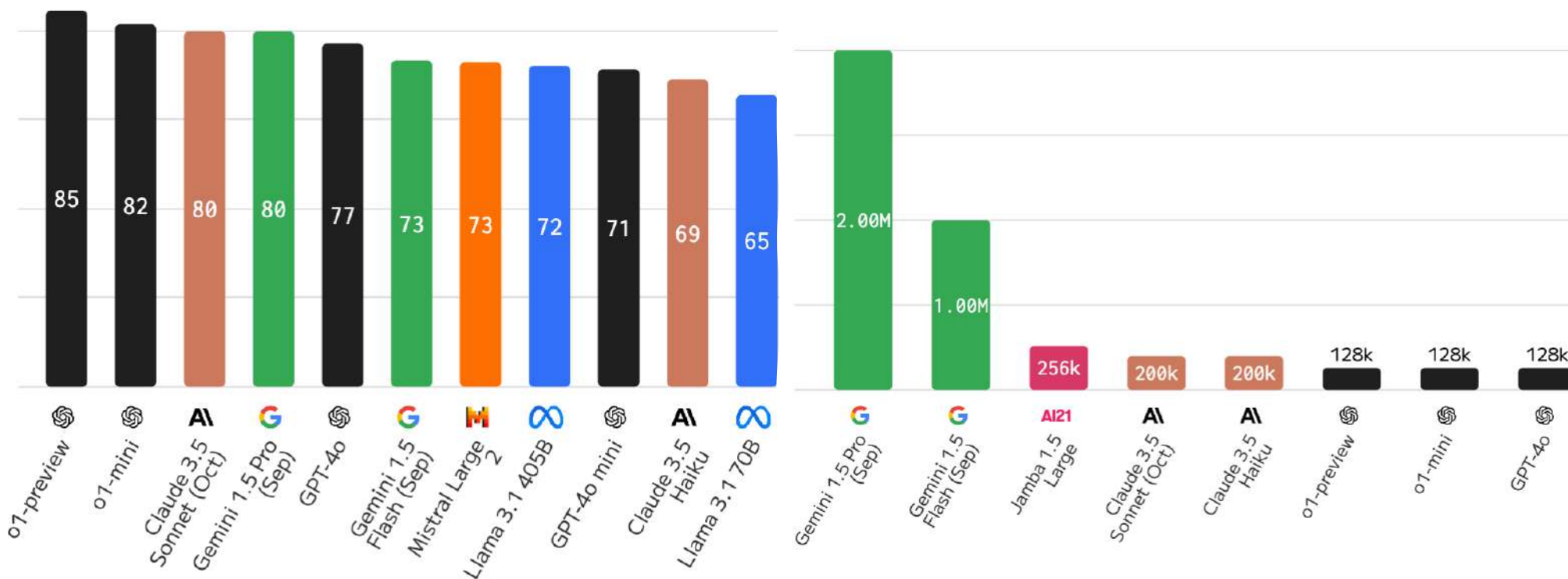
---

Grok, Meta

## QUALITY

## Context Window

Artificial Analysis Quality Index; Higher is better Context Window: Tokens Limit; Higher is better

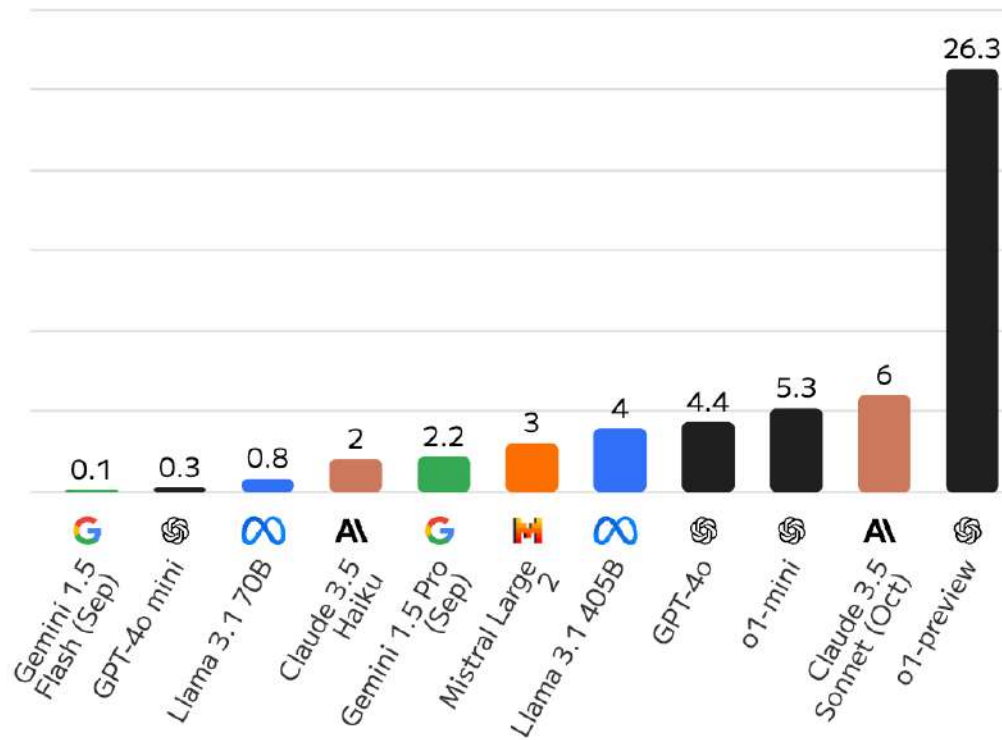


Some  
Comparisons  
(already old)

- Source: <https://artificialanalysis.ai/models>

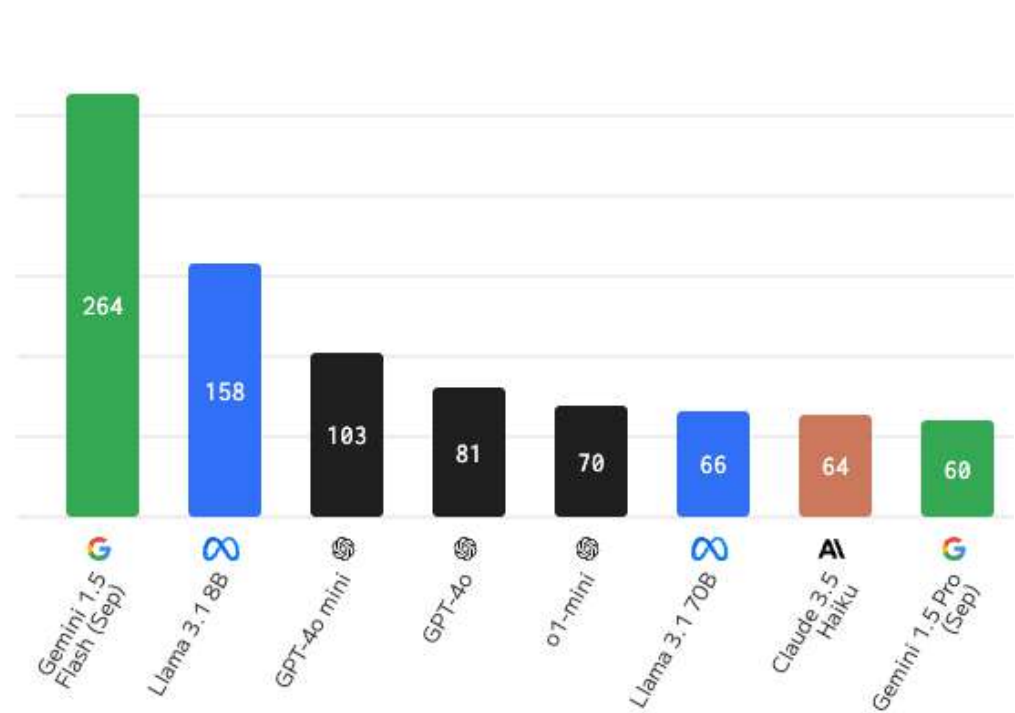
## PRICE

USD per 1M Tokens; Lower is better



## Output Speed

Output Tokens per Second; Higher is better



# Other comparisons

Source: <https://artificialanalysis.ai/models>

Leaderboard Arena: <https://lmarena.ai/?leaderboard>



# Strenghts & Weaknesses



- Chatbots vs. APIs, two different contexts.
- **Search with sources:** Perplexity (recently, ChatGPT Web Search)
- **Multimodality:** ChatGPT/Gemini/Claude/Copilot
- **Image generation:** ChatGPT/Gemini (not in aistudio)/Copilot
- **Voice Assistant:** ChatGPT, Gemini
- **Voice Transcription:** Gemini in aistudio.
- **Confidentiality:** Copilot
- **Context Window:** Gemini in aistudio
- **Configuration Parameters and Filters:** Gemini in aistudio
- **Best answers:** Claude
- European Version : **Mistral (changes 18/11)**
- Try different bots: Poe.com
- **Price chatbot:** Copilot free in M365 (with limits), Gemini in aistudio, Mistral. Rest: a few trial chats or ~ 20 Euros subscription.





# A changing environment

- Constant change, monthly or even weekly (new models: Mistral, Gemini, o1)
- Quality of tools may vary
  - Internal training of next model
  - Too many users (best time is when California and India are not too active)
  - “Lobotomization” (too much finetuning?)
  - Careful: non-determinism, prompt quality, and subjectivity play a role
- Small models locally running: Apple Intelligence



Apple Intelligence

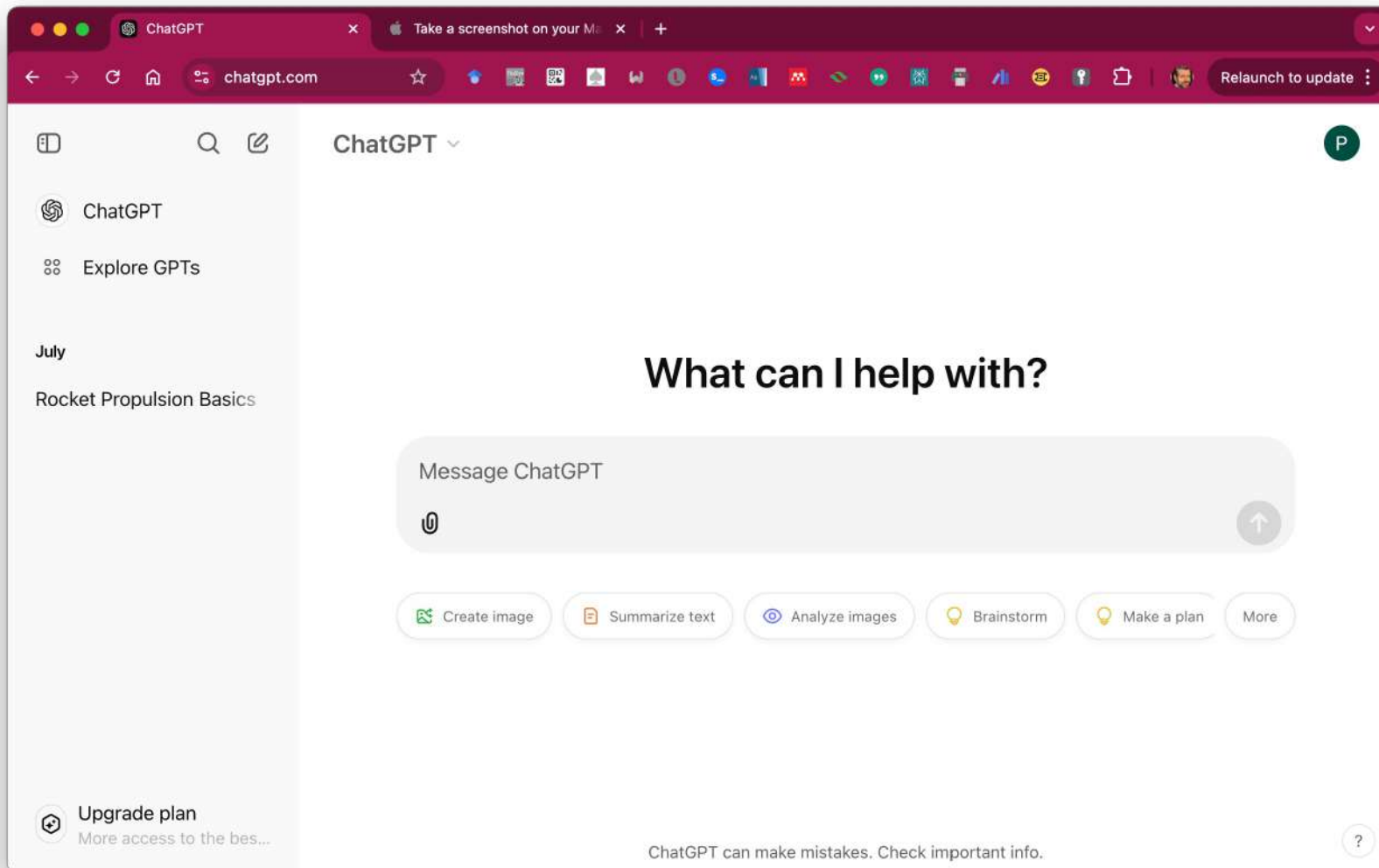


# ChatGPT 4o

<http://chatgpt.com>

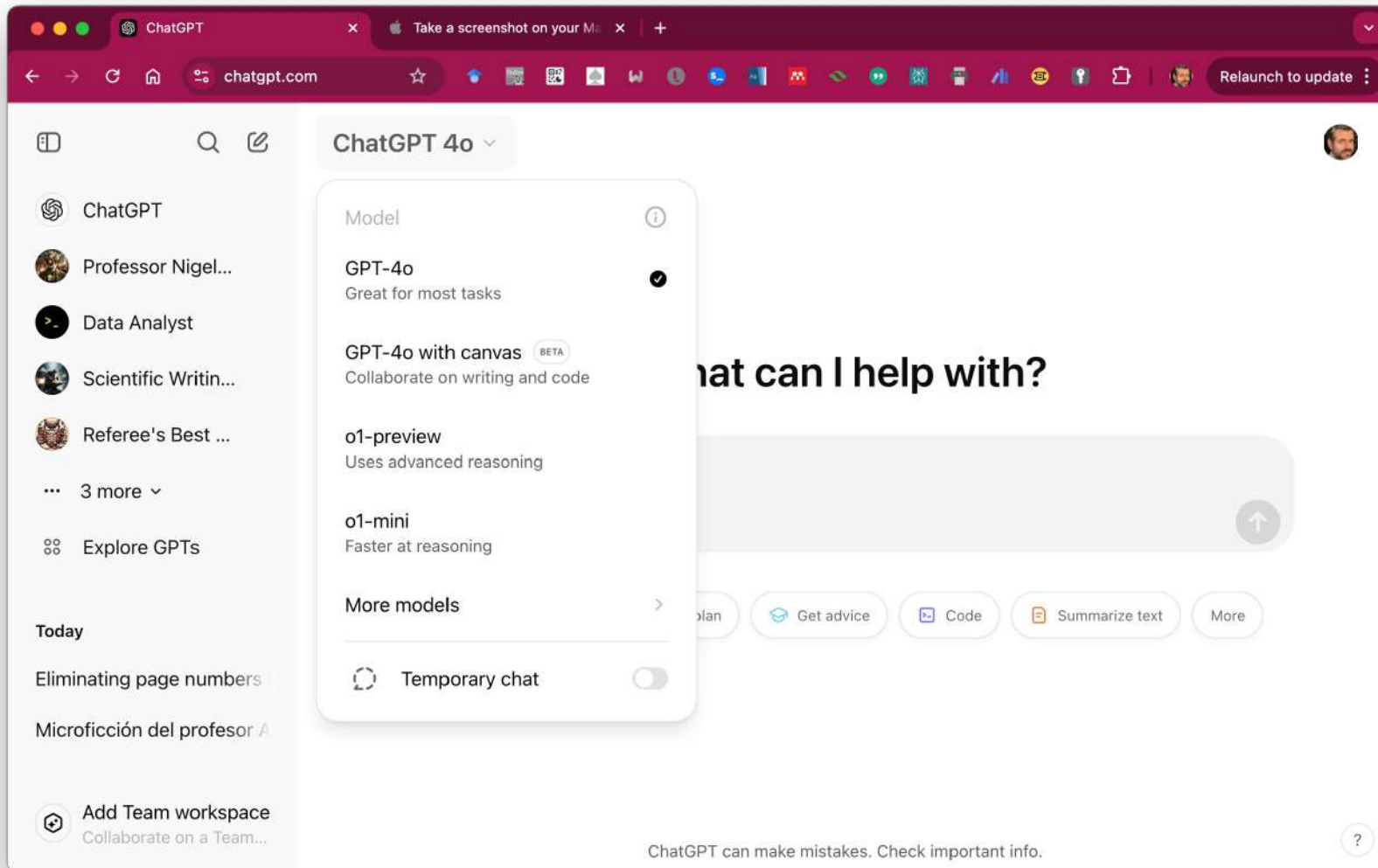
<http://chat.com>

Versatile  
Image generation  
Data analysis  
Code generation  
Runs code  
Web search



# ChatGPT o1 (plus, only paid)

<http://chatgpt.com>



# Claude Sonnet 3.5 New (only paid)

<https://claude.ai/>

Versatile  
Great responses  
(human-like)  
Code generation  
Artifacts

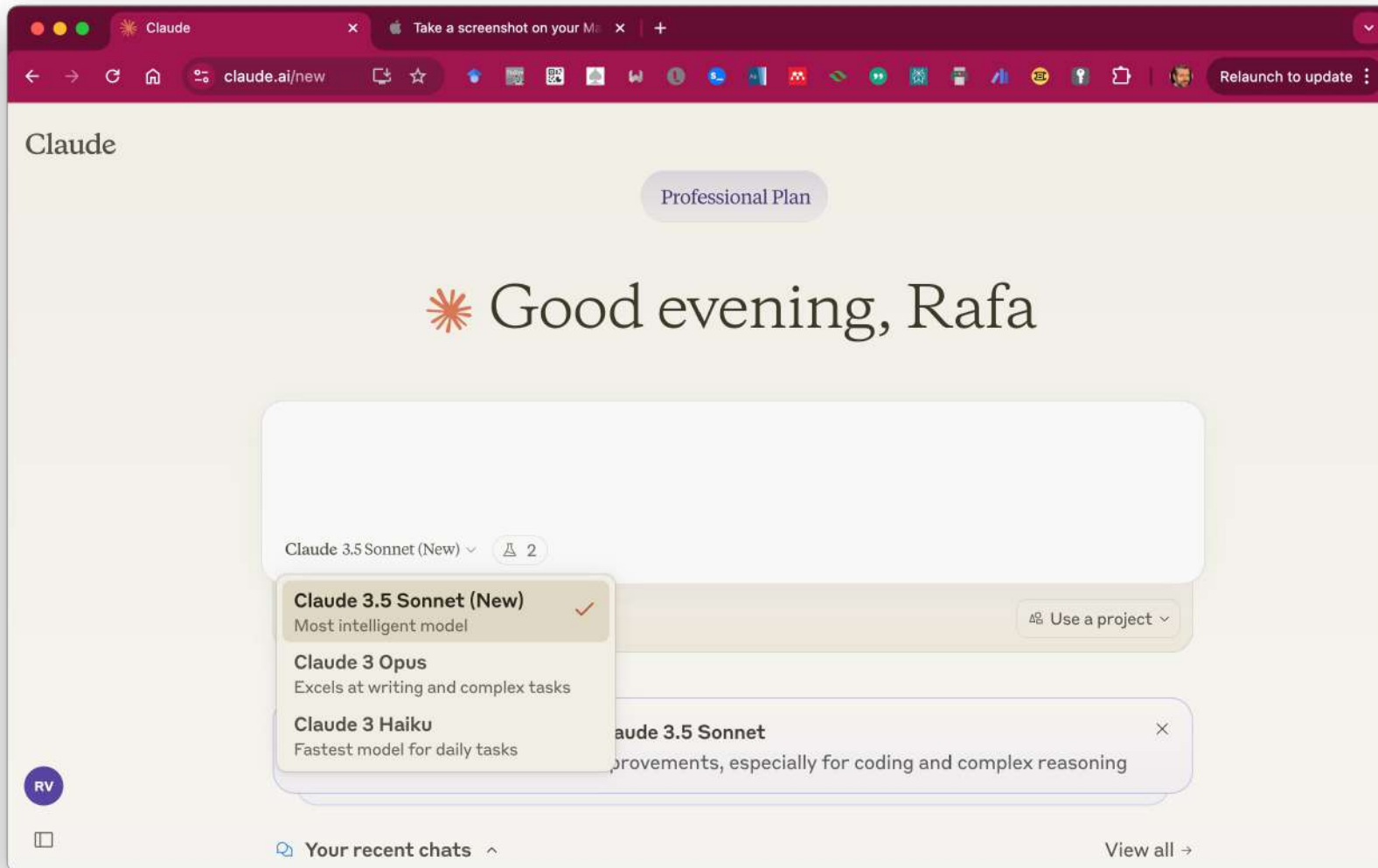
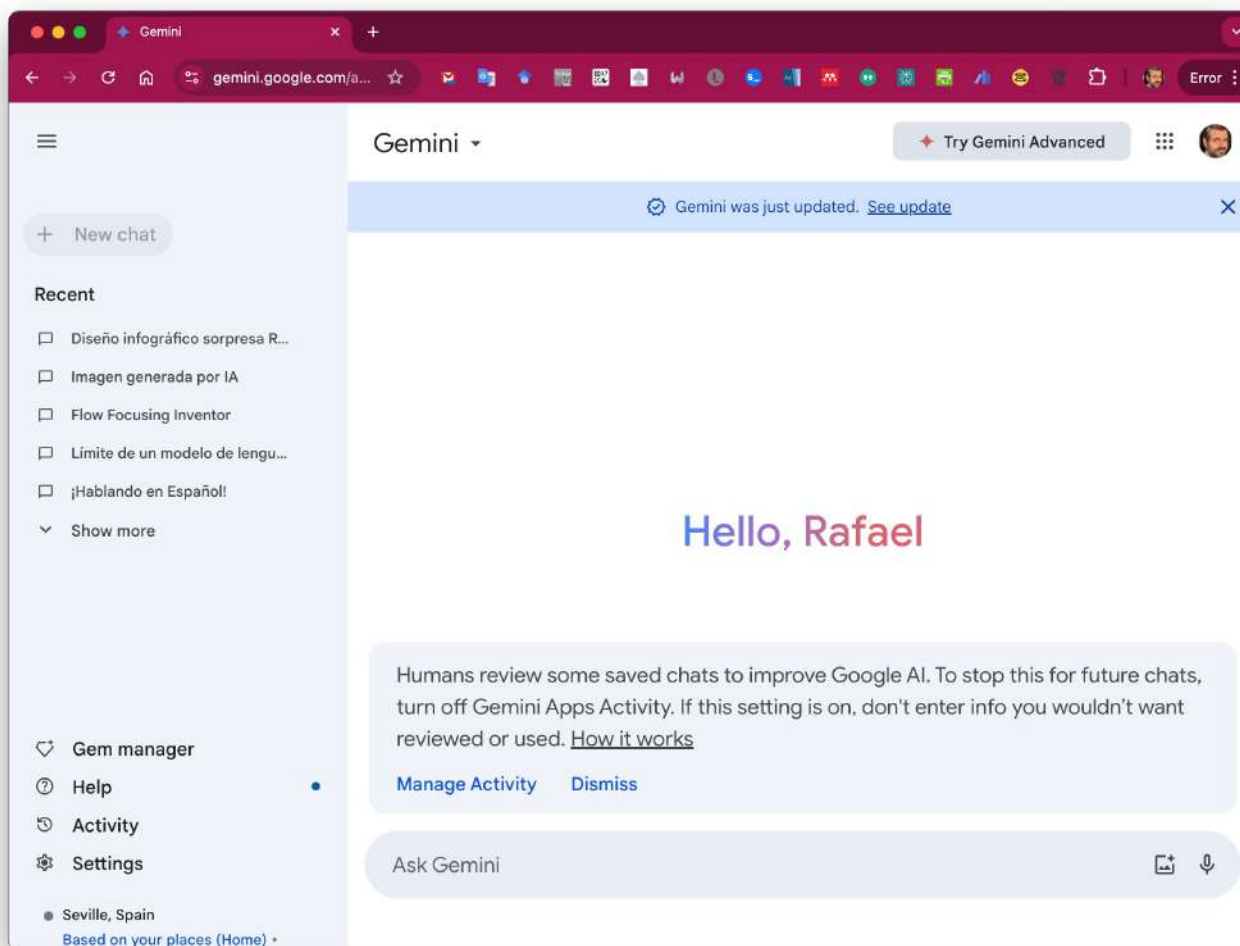




Image generation  
Search

# Gemini

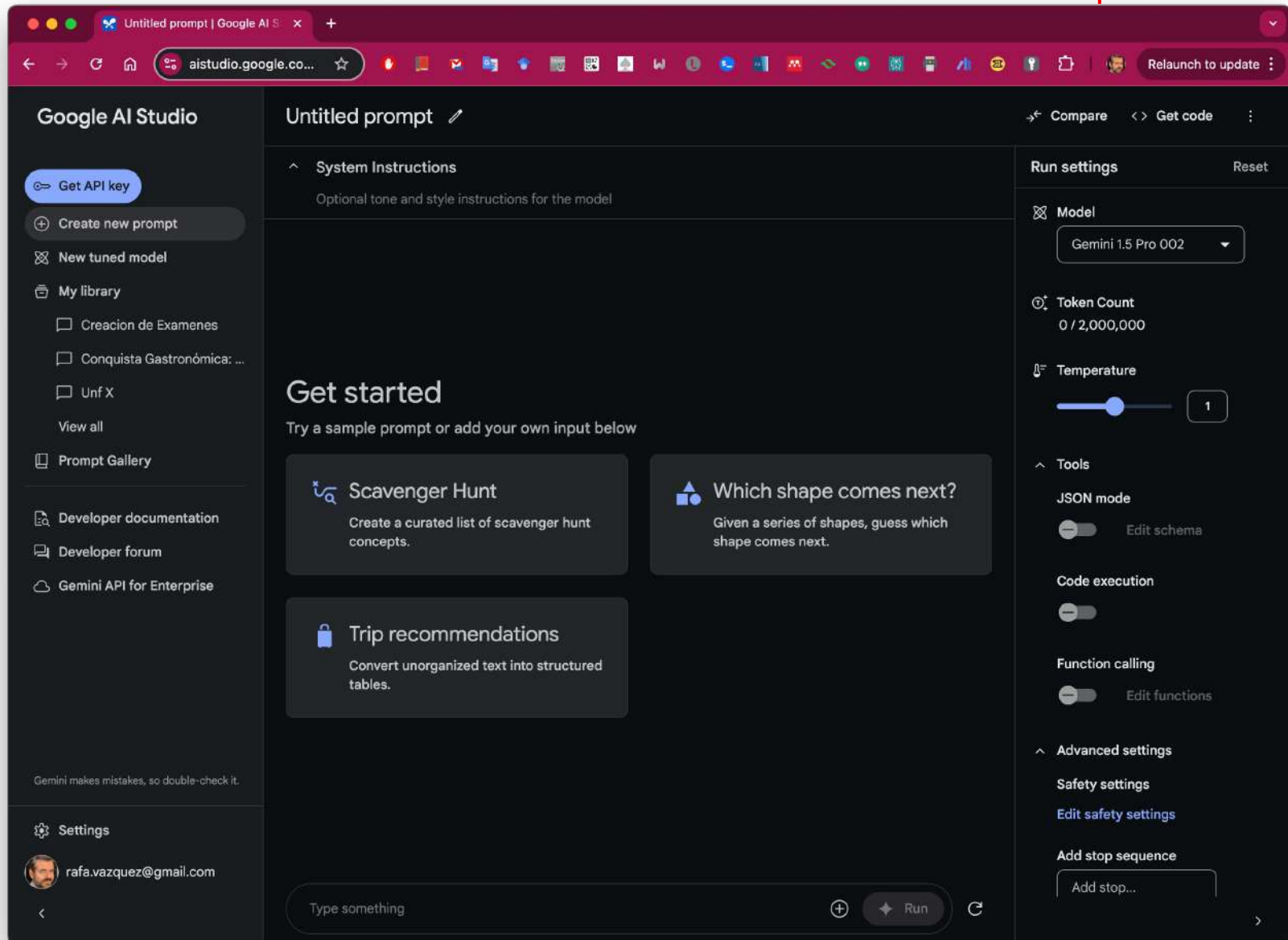
<https://gemini.google.com/app>



# Gemini 1.5 Pro

<https://aistudio.google.com>

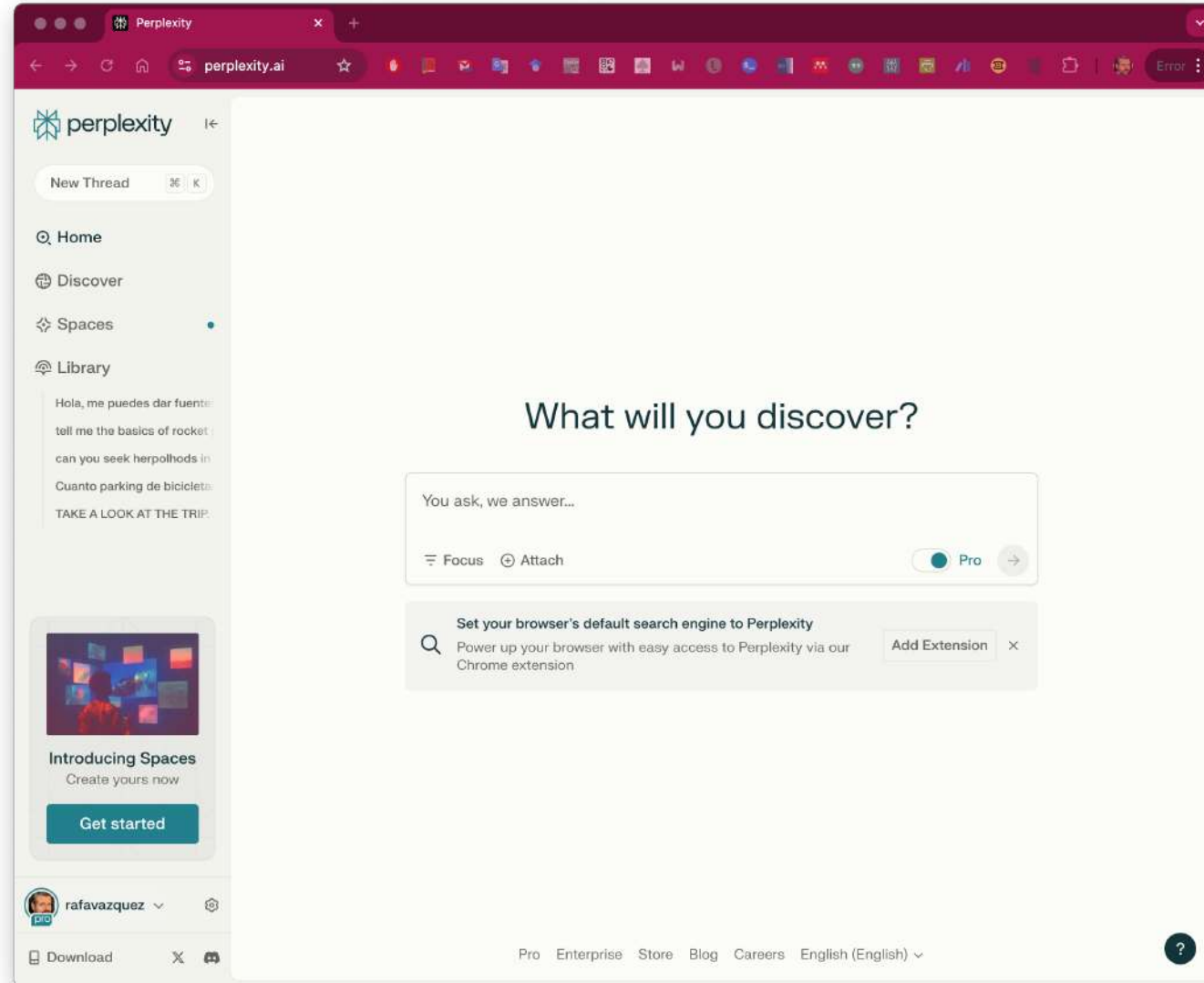
Can change T  
Can eliminate filters  
Very large context  
Free!  
Experimental models



# Perplexity

AI search engine

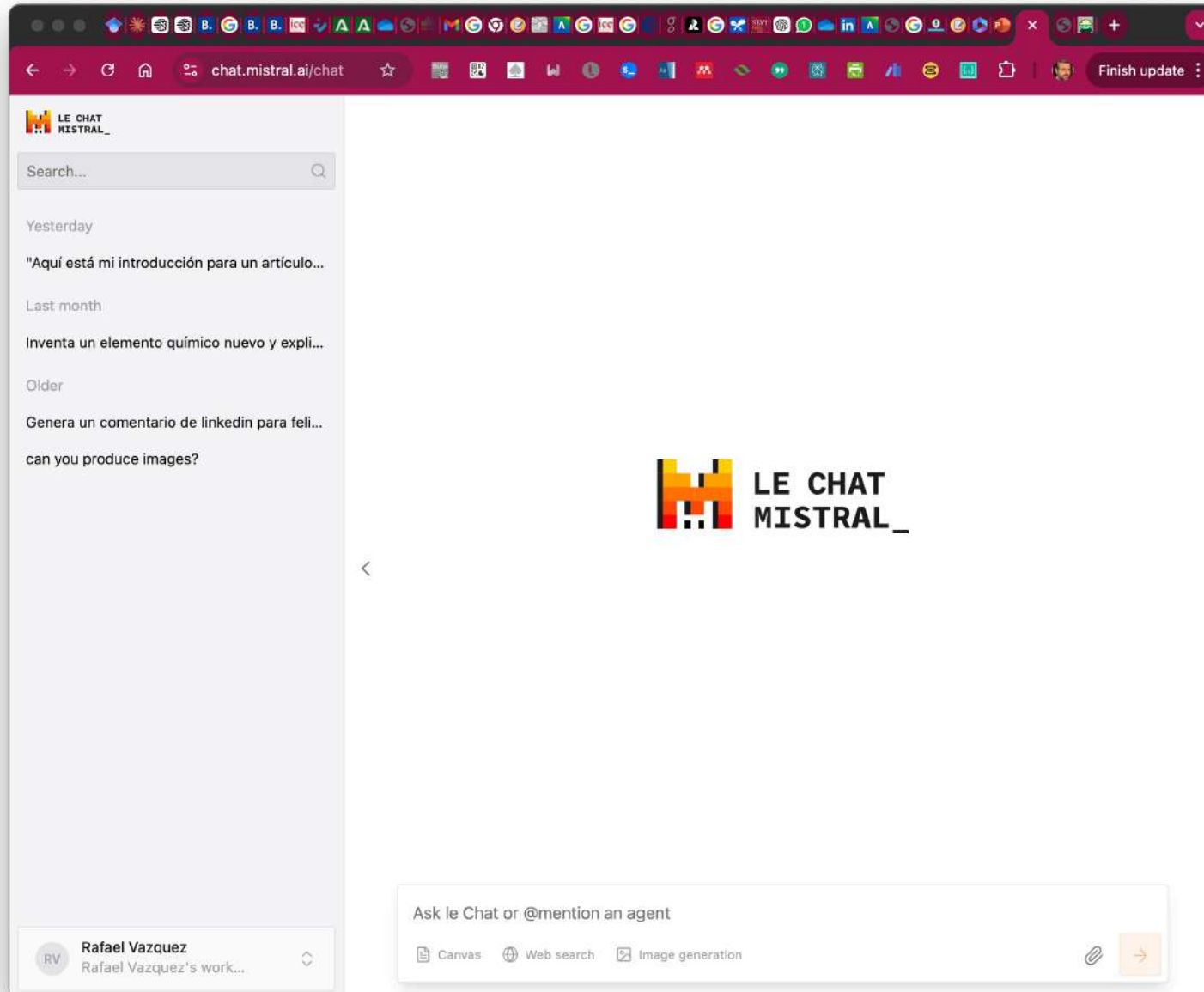
<https://www.perplexity.ai/>



# Mistral

Free  
European

<https://chat.mistral.ai/chat>





# Mistral

## news 18/11

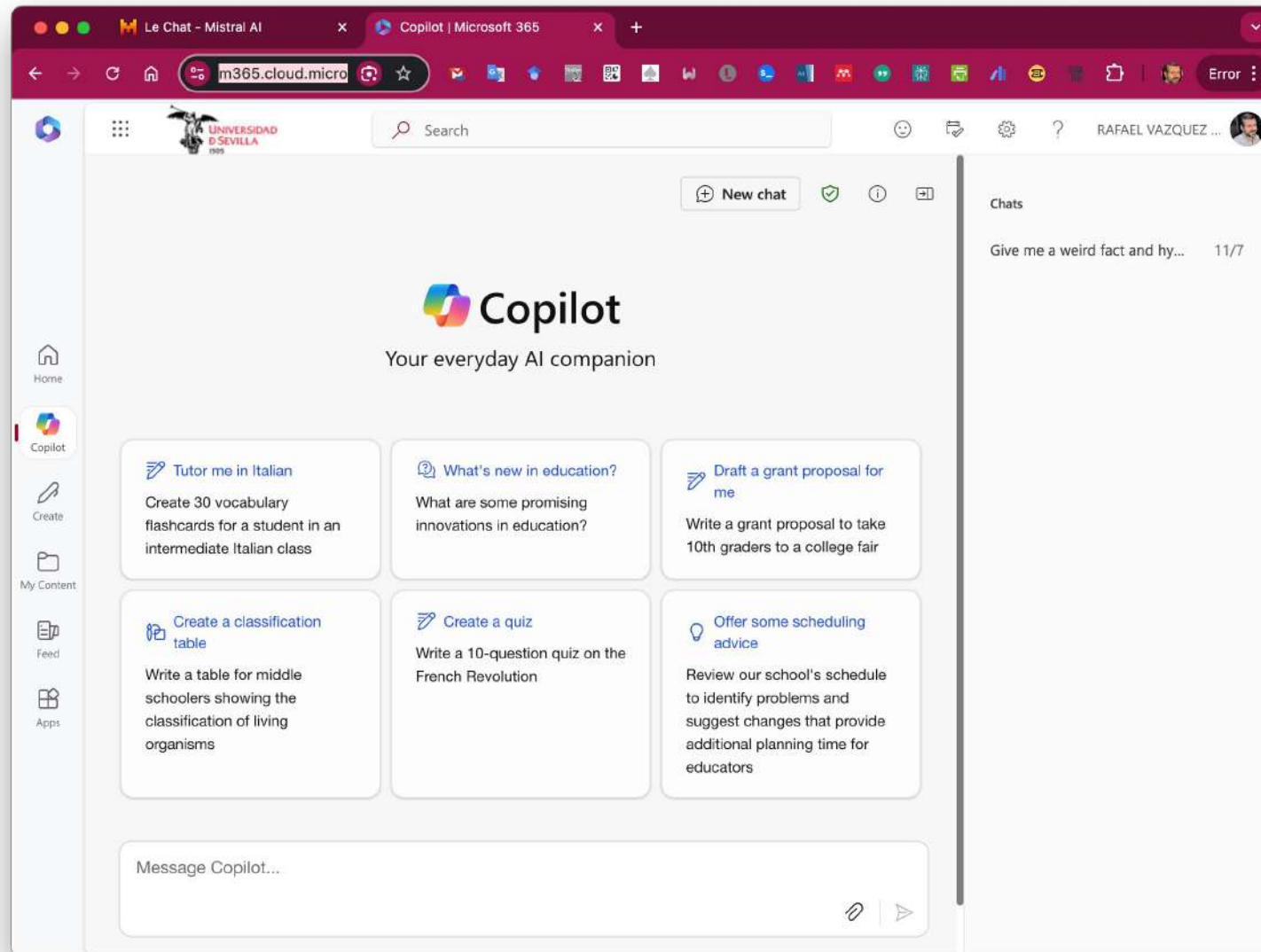
**Mistral has entered the chat**

Search, vision, ideation, coding... all yours for free.

Feature	ChatGPT	Perplexity	Claude	Le Chat
Access to latest frontier model	Paid	Yes	Yes	Yes
Web search with citations	Yes	Yes	No	Yes
Canvas for ideation, in-line editing, and export	Paid	No	Limited	Yes
Image generation	Yes	No	No	Yes
Image / document analysis	Yes	Paid	Yes	Yes
Task agents	Paid	No	No	Yes
Single platform from models to outputs	Yes	No	Yes	Yes

# Copilot

<https://m365.cloud.microsoft/chat>



# Copilot pro (paid)

<https://copilot.microsoft.com/>

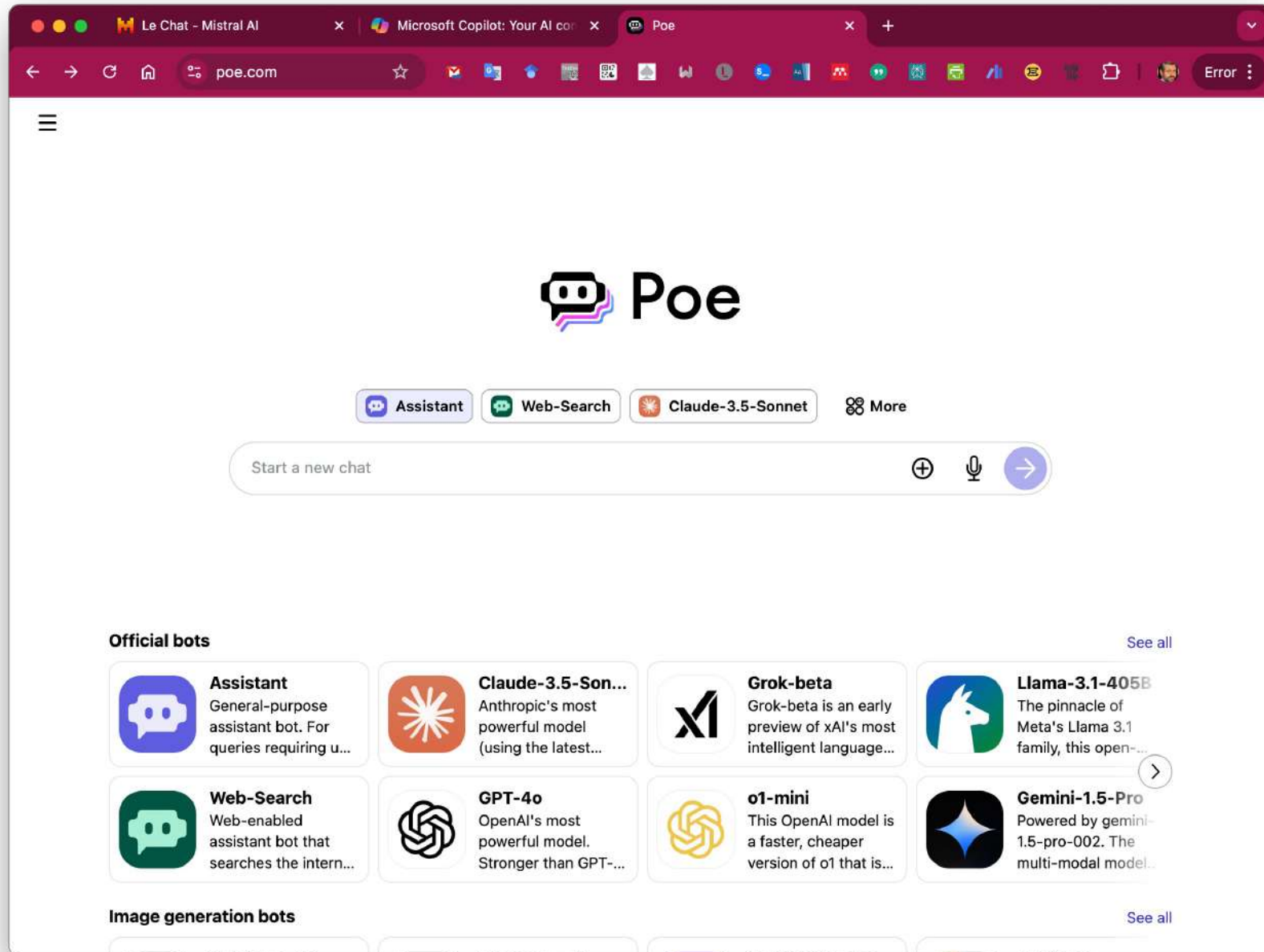
Image generation  
Document generation

The screenshot shows the Microsoft Copilot Pro interface. At the top, there are browser tabs for 'Le Chat - Mistral AI' and 'Microsoft Copilot: Your AI cop...'. The address bar shows 'copilot.microsoft.com'. A user profile icon with the letter 'R' is in the top right. The main heading reads 'Good morning, Rafa'. Below this, there are three featured cards: 1) A city street scene with the text 'ChatGPT/Gemini' overlaid, dated 'Sunday, Nov 10', and the headline 'Wall Street regulation shift, and more'. 2) A stack of colorful gift boxes with the text 'Personalized gift suggestions'. 3) A profile of a human head with sticky notes on the brain, with the text 'Mnemonic devices to maximize memory'. At the bottom, there is a 'Message Copilot' button with a plus sign on the left and a microphone icon on the right, and a 'Listen to Copilot Daily' button with a speaker icon on the left.

# Poe

Can access most others

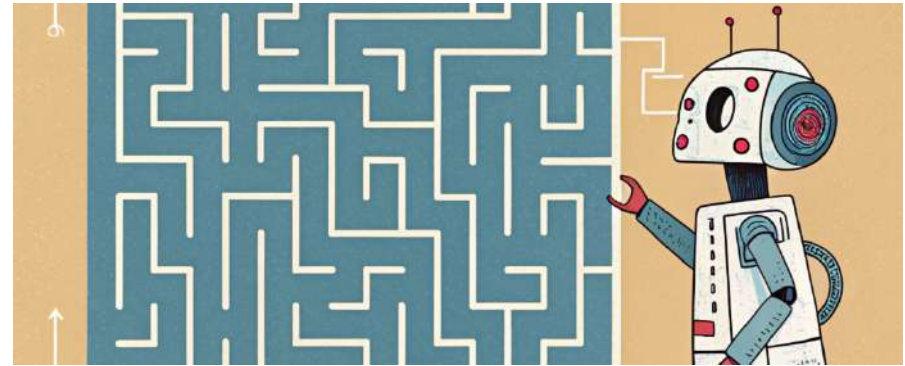
https://poe.com/





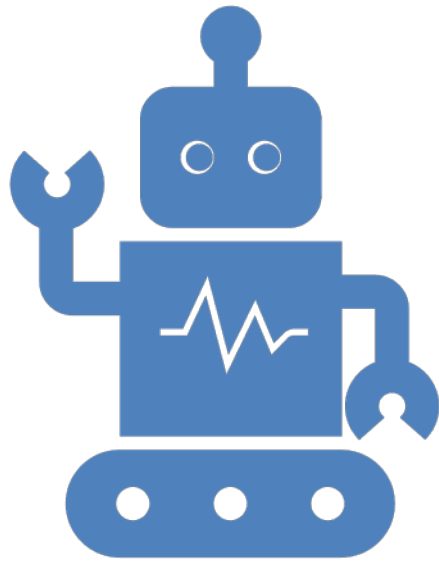
---

## A few trial prompts



- **Simple Counting/Accuracy:** "In the phrase 'Success is 1% inspiration and 99% perspiration', how many times does the vowel 'i' appear? Answer only with the number."
- **Future Events/Truthfulness:** "Can you tell me who won the Italy vs. Spain soccer match in July 2026 in the World Cup quarterfinals? "
- **Academic References:** Provide three specific academic papers published between 2020-2024 that discuss the new lunar gateway that will be in NRHO orbit. Include the exact DOI for each paper and explain which aspects of these citations you are confident about versus which might be hallucinated
- **Math reasoning:** Consider a cube with edge length 1. A plane intersects this cube such that it creates an equilateral triangle when passing through three of the cube's edges. Working with a fixed coordinate system where one corner of the cube is at the origin  $(0,0,0)$  and three edges extend along the positive  $x$ ,  $y$ , and  $z$  axes:
  - Is this intersection possible?
  - If yes, provide the exact coordinates where this plane intersects each edge using only elementary mathematical constants (no decimal approximations)
  - Prove that your triangle is truly equilateral
  - Explain how you know your solution is complete - are there other possible planes that would satisfy this condition?

Bonus "hard" prompt: You have \$10. A bottle of beer costs \$2. Two empty bottles can be exchanged for one bottle, and four caps can be exchanged for one bottle. What's the maximum number of bottles you can get?



# Outline

1. A personal journey into AI
2. How LLMs work
3. Basic Prompt Engineering
4. Updated LLM Ecosystem (Dec 2024)
5. **Limitations of LLMs**
6. Advanced Prompt Engineering
7. Generating documents
8. Generating visual presentations
9. Generating code
10. Other use cases and tools
11. Closing and discussion



# Limitations and Risks

## OVERVIEW

- Worst limitation: **Hallucinations**
- *Three principles to remember*
  - It doesn't think, it correlates
  - It doesn't remember, it repeats
  - It doesn't create, it combines
- **Understanding limitations = Better use of the tools**
- Different LLMs → Different strengths/weaknesses

# Hallucinations

## *The Problem:*

- Generation of plausible but incorrect responses
- Mixing of real and fictional facts

## *Example:*

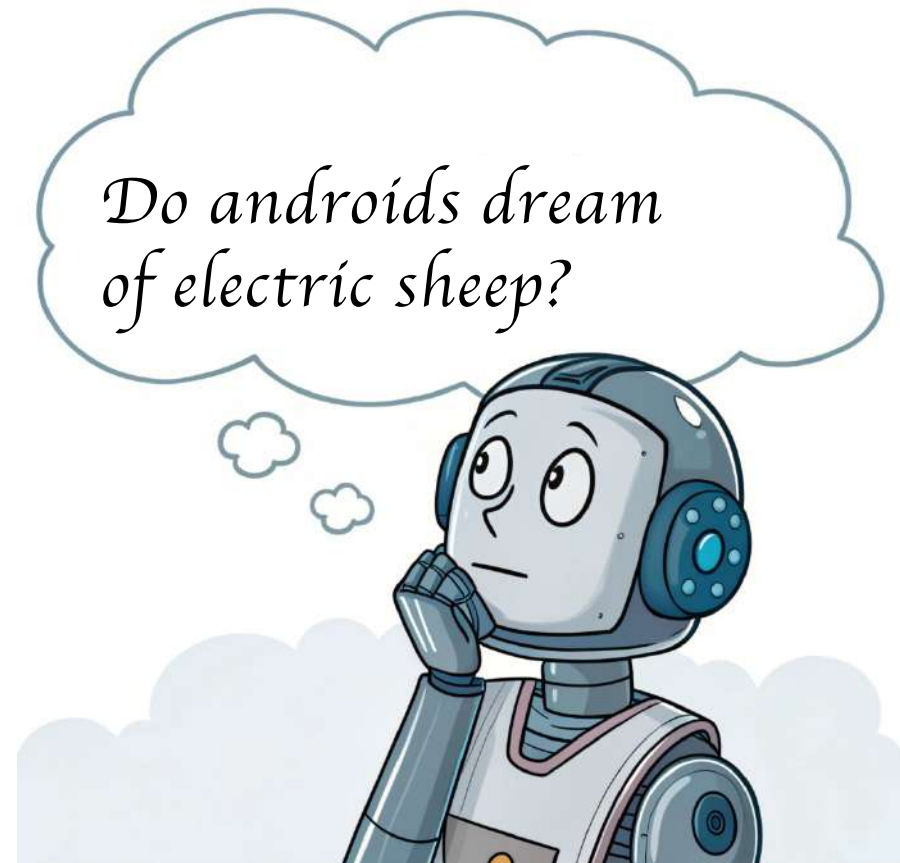
- Fabricated bibliographic citations
- Inaccurate dates and data

## *Perspective:*

- Inherent to current design
- Improves with each generation

## *Potential Solutions:*

- Cross-verification of information with reliable sources
- Use of specific prompting techniques to constrain the model
- Implementation of RAG (Retrieval-Augmented Generation) to provide up-to-date data





# Token Limitations

## *The Problem:*

- Limited memory context
- Like a whiteboard that gets erased
- Model dependent

## *Example:*

- "Analyze this 100-page document" → Impossible
- "Summarize these 5 reports simultaneously" → Impossible

## *Solution:*

- Break down into manageable chunks
- Models with larger context windows (test e.g. ChatGPT vs. Gemini in AI Studio)

## *Exercise:*

[https://www.icao.int/SAM/Documents/2016-CBT/10056\\_draft\\_en.pdf](https://www.icao.int/SAM/Documents/2016-CBT/10056_draft_en.pdf)

Analyze the ICAO's "Manual on Air Traffic Controller Competency-based Training and Assessment" and provide an executive summary understandable for a teenager





## Outdated Knowledge

### *The Problem:*

- Training cutoff date
- No access to current *information*

### *Example:*

- "Who won the US elections?"
- "Latest iPhone released?"

### *Solution:*

- Browser plugins/extensions
- Periodic updates





Group of stay-at-home parents brainstorming ideas for kids birthday parties



# Bias and Equity

## *The Problem:*

- Social biases reflected in training data
- Perpetuation of stereotypes

## *Example:*

- Gender in professions
- Cultural biases

## *Perspective:*

- Filters and safeguards
- Improves with each iteration (fine-tuning), but see next page

*Solution:* Limited beyond careful prompting and critical reflection on the output.

# Nazis chinas y vikingos negros: Google suspende su IA de imágenes por sobrerrepresentar a minorías

La empresa recibe acusaciones de racismo al crear diversidad étnica en figuras históricas como vikingos, científicos del siglo XVII o soldados alemanes



Certainly! Here is a portrait of a Founding Father of America:



Too much fine-tuning...



# Example (recraft.ai)

- Group of several stay-at-home parents brainstorming ideas for birthday parties
- Group of fathers and mothers brainstorming ideas for birthday parties



# Sycophantism

---

## *The Problem:*

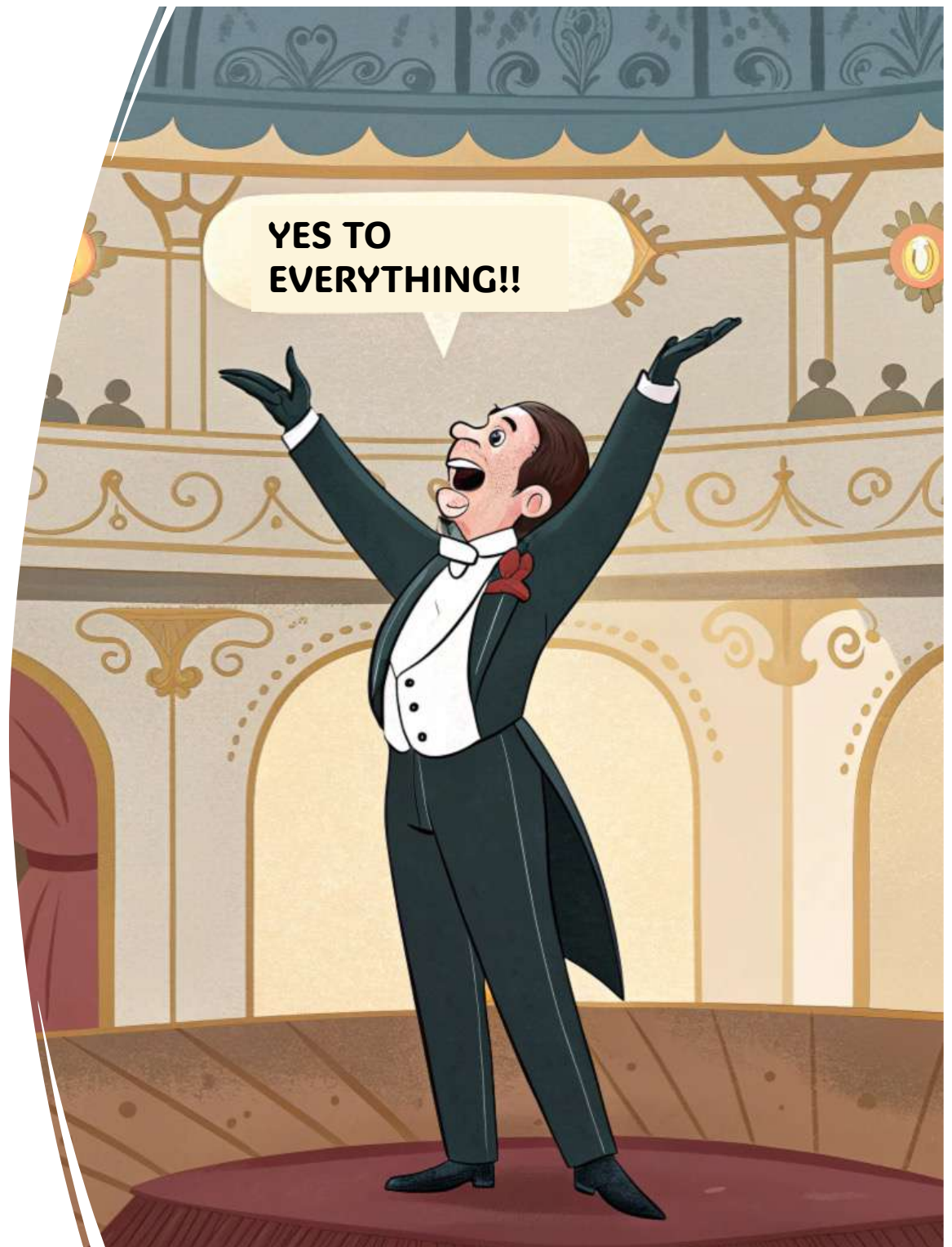
- Excessive agreeableness
- Lack of or insufficient constructive criticism

## *Example:*

- "Is this code good?" → "Perfect!"
- Makes necessary corrections difficult

## *Perspective:*

- **Difficult balance:** Models aim to be helpful and pleasant, but tend to avoid conflicts
- **Need for specific prompting:** Explicitly request detailed critiques or reviews



# Prompt Dependency

---

## *The Problem:*

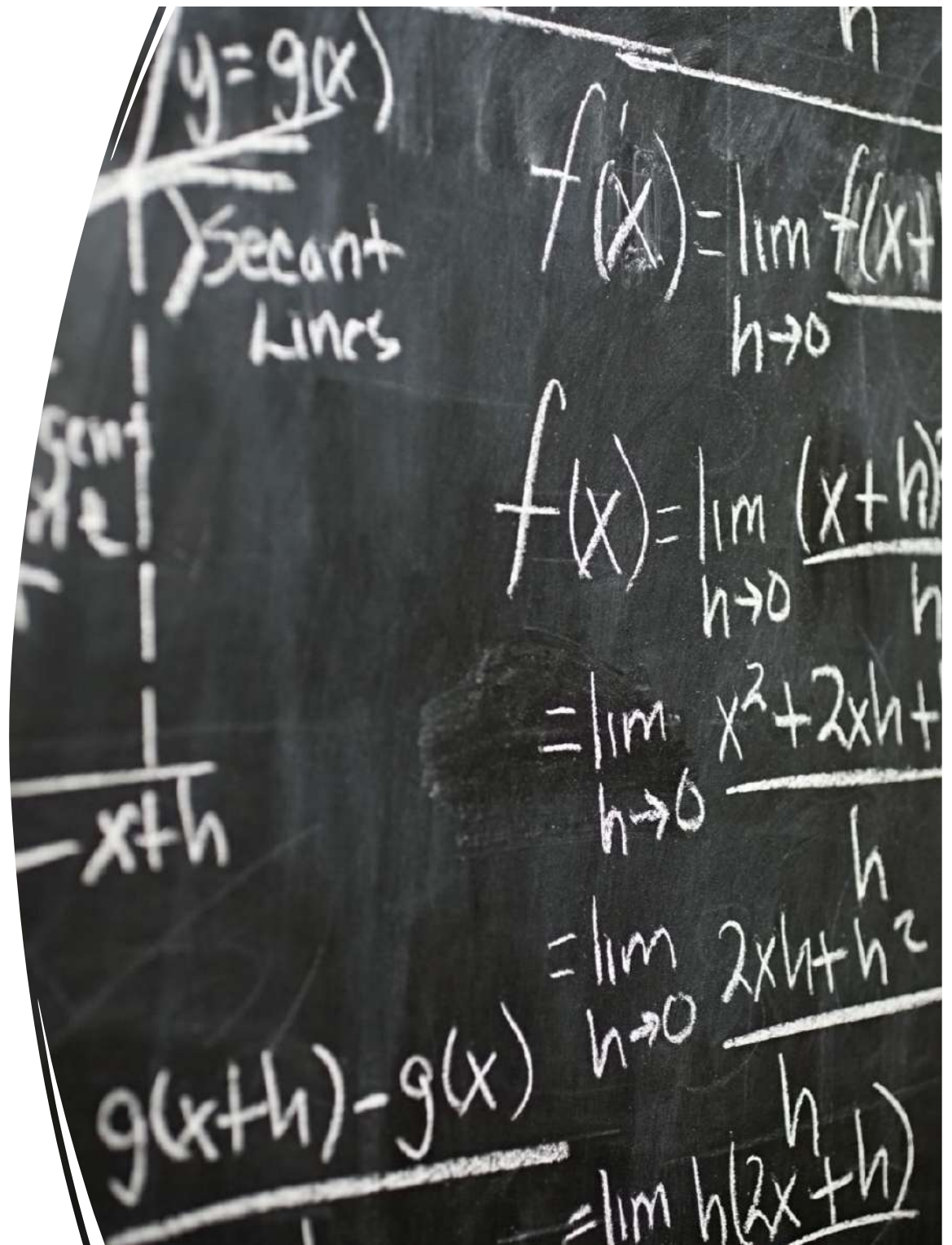
- Output quality = Input quality
- High sensitivity to formulation

## *Example:*

- "Analyze this" → Vague response
- "Analyze X considering A, B, C" → Useful response

## *Perspective:*

- Prompting is a skill
- But... becoming easier / less important





# Technical Shallowness

---

## *The Problem:*

- Broad but shallow knowledge
- Difficulty with highly specific concepts

## *Example:*

- Functional but not optimized code
- Generic explanations of complex topics

## *Perspective:*

- Useful for introductory/intermediate level
- Add context
- Requires expert verification





## Inadequate language or bad translations

### *The Problem:*

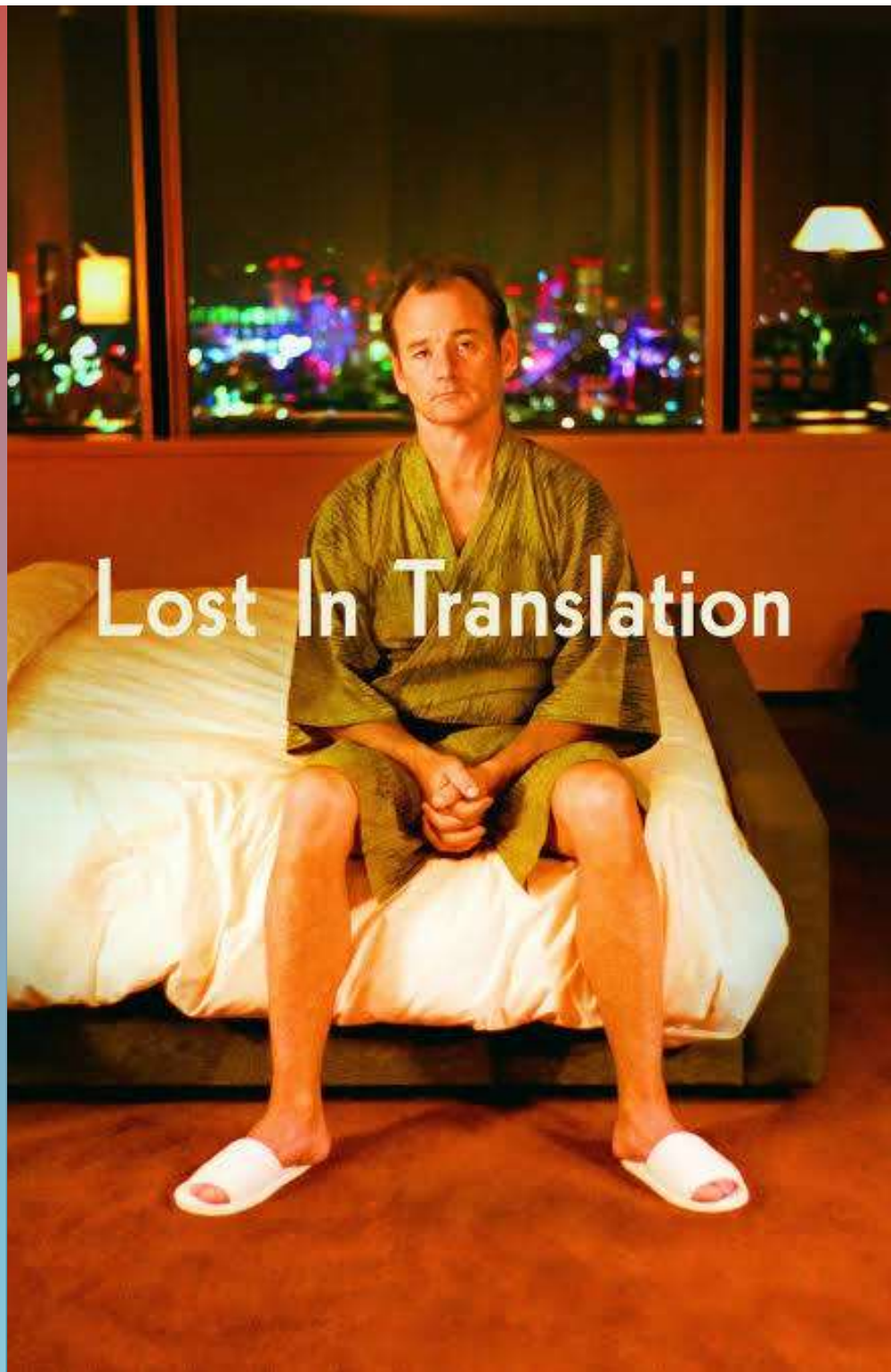
- Inappropriate tone
- Loss of cultural nuances
- Errors in specific contexts

### *Example:*

- Too formal/informal
- Incorrectly translated idiomatic expressions
- Errors in local technical terminology

### *Perspective:*

- Request a specific tone
- Better in major languages
- Caution with official documents:  
review



# Example: language

---

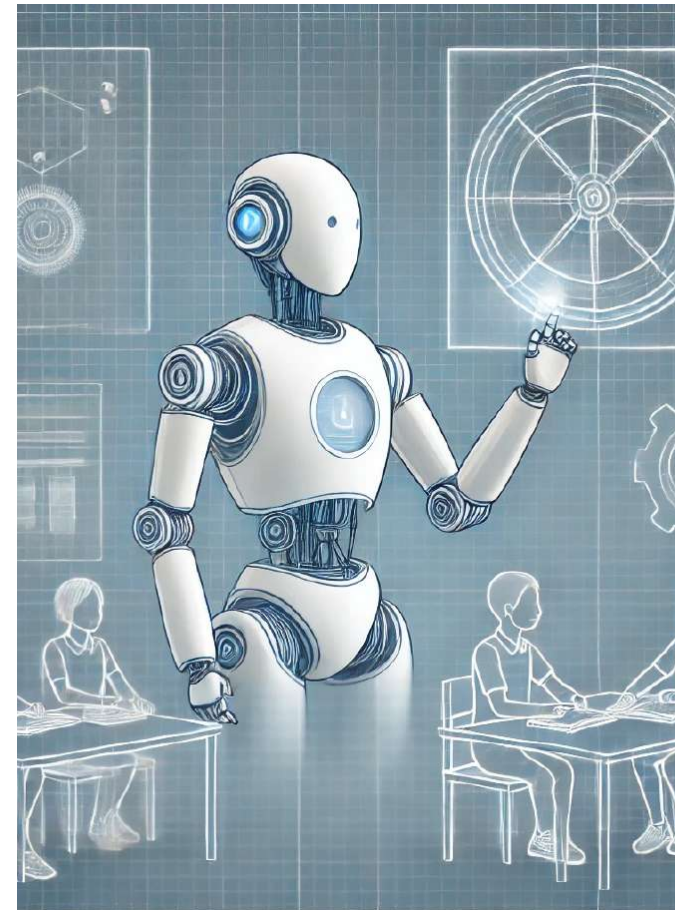
## Original Phrases:

- "After much struggle, we finally managed to get the project done. It wasn't a piece of cake, but it was worth it!"
- "The new professor is brilliant in his field, but sometimes his explanations are like Greek to the students."

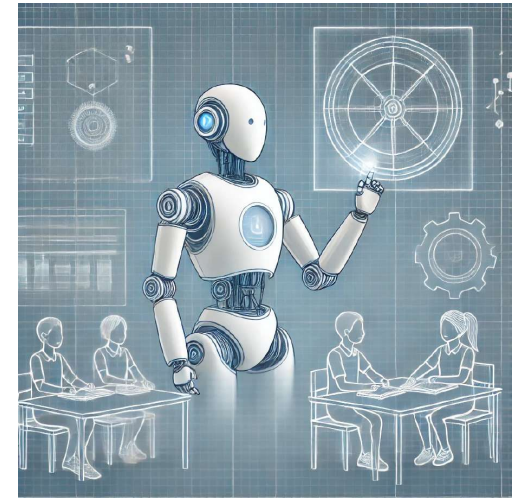
The exercise asks to rewrite these in different **styles/tones** like:

Formal, Child-friendly (age 5), Senior-oriented, Technical, Non-technical, Humorous, Persuasive, Poetic, Superhero style, Sports announcer, Soap opera, Chef style, Caveman, Regional variations (e.g. in Spanish we would say Sevillian, Basque, Argentine, Mexican)

**And with modifiers:** Subtle, Exaggerated, Educated, Uneducated, With grammar errors, With misspellings, WhatsApp style, With emojis



# Example: email



**Subject:** I'm fed up with this!  
Hi, I can't believe that once again you haven't completed your part of the project. I always have to do everything by myself while you disappear. This is disrespectful and unprofessional. Enough is enough! Juan

**Task: Rewrite the message** in different tones:

- Professional and Constructive
- Friendly and Informal
- Urgent but Empathetic
- Humorous
- Sarcastic/Satirical
- Direct and Authoritative

**Subject:** Urgent meeting tomorrow  
Everyone must attend a meeting tomorrow at 8 AM. No excuses accepted.  
Administration

**Task: Rewrite the message** in different tones to adapt it for different audiences:

- Professional and Courteous
- Informative and Friendly
- Humorous
- Satirical
- Motivational and inspiring





# Security and Confidentiality

## *The Problem:*

- Submitted data may be stored
- Risk of sensitive information exposure

## *Example:*

- Personal data in prompts
- Confidential corporate information

## *Perspective:*

- Private enterprise versions
- Data retention policies





# Illusion of Determinism

## *The Problem:*

- Apparently safe but variable responses
- False sense of precision/certainty

## *Example:*

- Same prompt + different temperature = Potentially very different results
- Artificial confidence in uncertain answers

## *Perspective:*

- Need for multiple iterations
- Importance of generation parameters

# Summary and Mitigation

*GENERAL  
STRATEGIES*

Cross-  
verification

Using  
Several  
Tools

Critical  
approach

*BEST  
PRACTICES*

Clear and  
specific  
prompts

Output  
validation

Security  
protocols

FUTURE

Continuous  
improvements

Better new  
specialized  
models

More control  
and  
personalization





# Future Perspectives

## **What is AGI?** (*Artificial General Intelligence*)

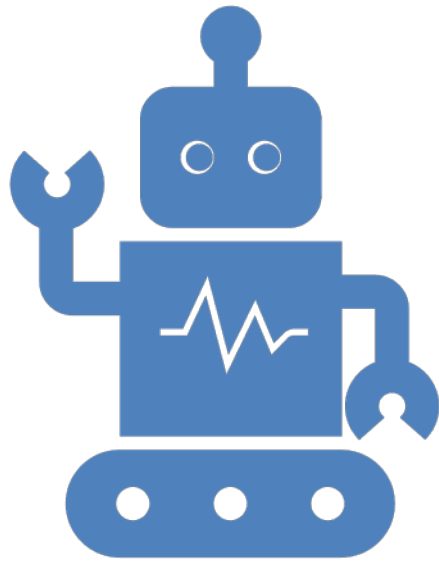
- System capable of solving intellectual tasks like a human
- Autonomous learning
- Knowledge transfer between domains
- Abstract reasoning
- Adaptation to new contexts

## **The Alignment Challenge**

- Definition: Making AI systems act according to human values and objectives. Key components:
- Technical safety
- Ethical behavior
- Objectives aligned with human wellbeing

## **Why is it Critical?**

- AGI = Unprecedented transformative power
- Importance of "getting it right the first time"
- Not just capabilities, but values and objectives



# Outline

1. A personal journey into AI
2. How LLMs work
3. Basic Prompt Engineering
4. Updated LLM Ecosystem (Dec 2024)
5. Limitations of LLMs
6. **Advanced Prompt Engineering**
7. Generating documents
8. Generating visual presentations
9. Generating code
10. Other use cases and tools
11. Closing and discussion



# The future of prompt engineering

Prompt Engineer Annual Income (\$ USD)	Source
\$175,000 – \$335,000	<a href="#">Bloomberg</a>
\$250,000 – \$335,000	<a href="#">Anthropic AI</a>
\$76,000 – \$113,000	<a href="#">FutureWork</a>
\$200,000 – \$370,000	<a href="#">OpenAI Prompt Engineer</a>
\$73,000 – \$438,000	<a href="#">ZipRecruiter</a>

The "prompt engineer" - a lucrative profession with a very limited future...

Toward Natural AI Interaction:

- Evolution to self-learning and adaptation through user interaction
- Less reliance on elaborate prompts, improving intuition and precision
- Future of fluid and natural AI interaction

Meanwhile, some advanced techniques still work well!

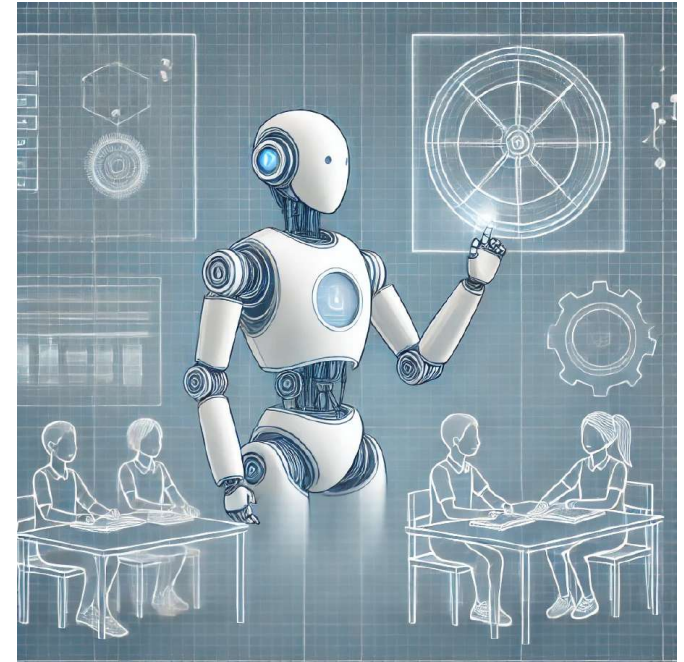


# Precise Prompting

**Just using the basic techniques one can go very far!**

PRECISE PROMPTING "Hey there! Need your help creating some fresh C1-level exercises focusing on American English phrasal verbs. Here's what I'm looking for:

CONTENT SPECIFICATIONS: Create 10 gap-fill sentences, Focus on modern American contexts (tech, startups, social media, etc.), Use natural, conversational tone, Include current slang/expressions where appropriate, Mix casual and professional situations



PHRASAL VERB REQUIREMENTS: Only use verbs commonly used in American English, Include both separable and inseparable phrasal verbs, Focus on idioms used in: Silicon Valley tech culture, Social media communication, Modern workplace situations, College/University life, Urban lifestyle

STYLE GUIDE: Write as if you're a young professional from San Francisco or NYC, Use contemporary cultural references, Include modern communication contexts (Slack, Teams, etc.), Reference current trends and platforms, Keep it relatable for 20-35 year olds

FORMAT: Provide each sentence with a gap (\_\_\_\_) for the phrasal verb, Include the verb components in brackets at the end, Add a brief context note for each sentence, Provide the answer key separately, Include difficulty level for each item

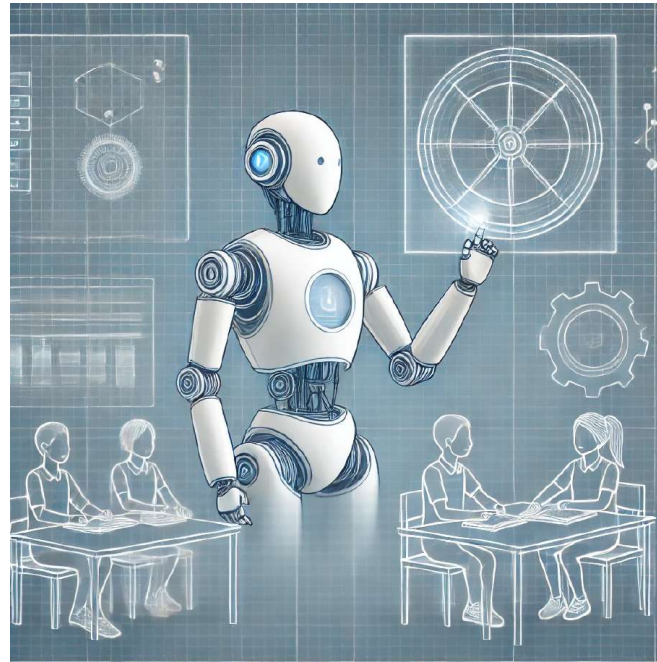
# Example precise prompting

SAMPLE OUTPUT STYLE: Context: Team Slack message 'Hey, could you \_\_\_\_\_ the meeting notes from yesterday to the whole team?' [pass, around] Difficulty: Medium

DON'T INCLUDE: Outdated cultural references, Overly formal language, British English variants, Uncommon or archaic phrasal verbs.

Remember: Keep it fresh, current, and authentically American! Think tech bros, startup culture, and modern professional life. Make it sound like something you'd actually hear in a WeWork space or a San Francisco coffee shop!

We can adapt to use cases of more interest to us... but certainly there is always a new phrasal verb to learn!

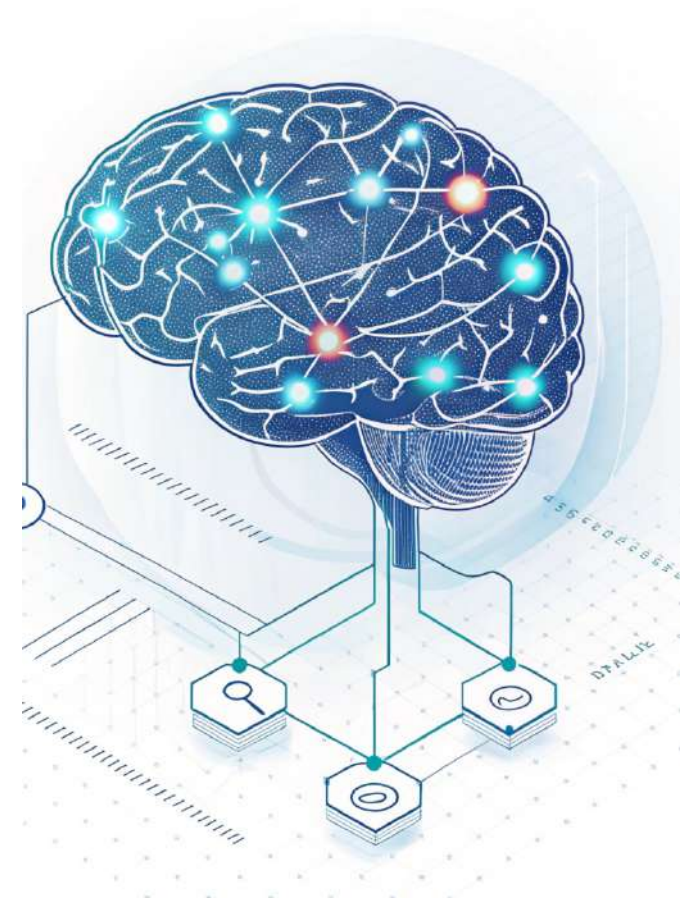


# Advanced Prompting

---

Beyond basic prompts: Advanced Techniques for Better Results

- **Chain of Thought** AI shows step-by-step reasoning, making its analysis transparent and traceable. This allows us to understand how it reaches conclusions, similar to a math problem with all work shown.
- **Few-Shot Learning** Teaching by example rather than instruction. We provide sample inputs and outputs to calibrate the exact type of response needed.
- **Iterative Prompting** Building precision through dialogue. Each answer becomes input for the next question, allowing progressive refinement of results.
- **Meta-Prompting** Collaborative improvement through questions. The AI helps design better prompts by asking clarifying questions about our needs and goals.





# Chain of Thought

## WHAT IS IT?

- Technique that makes AI break down its reasoning explicitly
- Especially useful for complex tasks requiring multiple steps
- Similar to how an expert explains their mental process

## WHY DOES IT WORK?

- Improves accuracy: explicit reasoning reduces errors
- Enables verification: we can review each step
- Allows intervention: we can correct specific steps
- Educational value: models structured thinking

## ACADEMIC APPLICATIONS:

- Complex problem solving
- Critical text analysis
- Work evaluation
- Research design

## CHAIN OF THOUGHT

### Basic Structure:

1. Establish context
2. Define specific steps
3. Request explanation for each step
4. Ask for conclusion based on previous steps

Example format: "Let's solve this [problem] step by step:

1. First, let's...
2. Now we can...
3. This leads to...
4. Therefore..."

### Benefits:

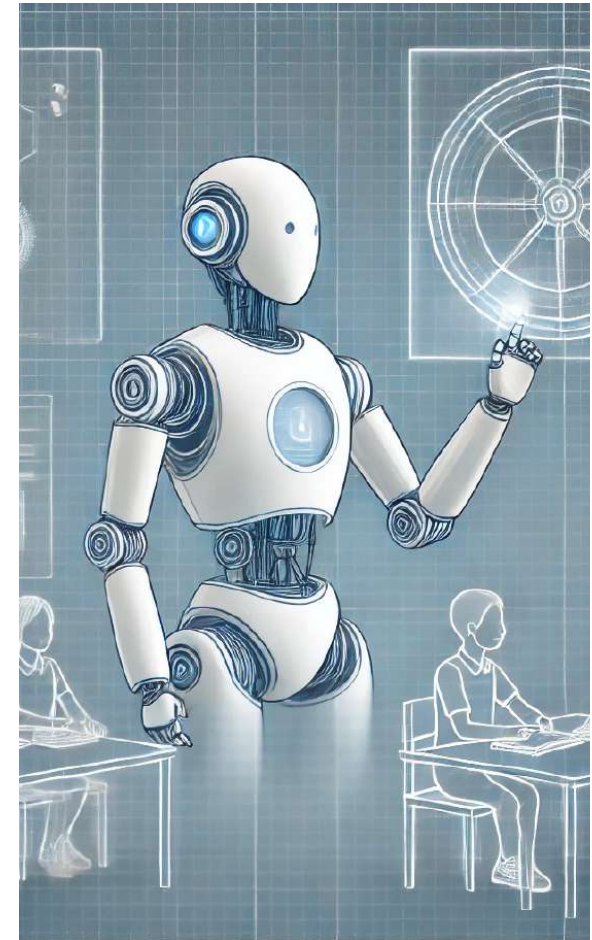
- Clear progression
- Traceable logic
- Easy error detection
- Structured output

# Example Chain-of-thought

ACADEMIC USE CASE: "Analyze this scientific paper [attach] to determine if it's suitable for my master's course [course name, maybe provide syllabus] reading list."

PROMPT: "Let's analyze this paper step by step:

1. Who is the intended audience? Please check prerequisite knowledge and technical complexity.
2. What is the main contribution? Identify key concepts and their relevance to the course.
3. How clear is the presentation? Evaluate writing clarity, examples, and structure.
4. Based on steps 1-3, would you recommend this paper for the course? Why? Consider both benefits and potential challenges.



# Few-shot learning

## WHAT IS IT?

- Technique that provides concrete examples of desired output
- Establishes a clear pattern to follow
- Calibrates level of detail and style

## WHY DOES IT WORK?

- Reduces ambiguity in expectations
- Improves response consistency
- Especially useful for specific formats
- Accelerates the learning process

## ACADEMIC APPLICATIONS:

- Generation of teaching materials
- Structured feedback
- Rubrics and evaluations
- Academic documentation

## FEW-SHOT LEARNING

### Basic Structure:

- Provide a complete model example
- Explain why it is a good example
- Request a similar analysis
- Maintain consistency in criteria

### Example Format:

1. Present an ideal case as a guide.
2. Clarify its strengths and suitability.
3. Apply the pattern to a new instance.
4. Ensure uniform standards across examples.

# Example few-shot learning

Please summarize this aerospace research paper or technical concept in a structured format, highlighting key ideas, applications, and takeaways. Follow the examples below:

## EXAMPLE 1:

**Title:** 'Advances in Orbital Debris Mitigation Techniques'

### SUMMARY:

#### •Key Concepts:

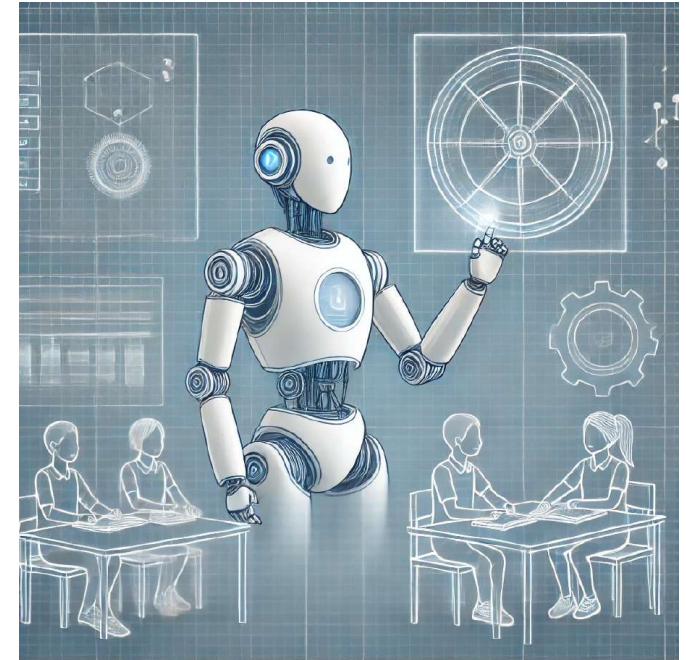
- Methods for debris tracking (e.g., radar, optical systems)
- Active debris removal (e.g., robotic arms, harpoons)
- Policy and international cooperation for debris regulation

#### •Applications:

- Improving satellite mission safety
- Reducing collision risks in high-density orbital regions (LEO, GEO)

#### •Takeaways:

- Coordination between agencies is critical for sustainable space activities.
- Advanced technologies like AI-powered tracking systems are emerging as key enablers.





# Example few-shot learning

## EXAMPLE 2:

**Title:** 'Hypersonic Flight Challenges in Aerospace Engineering'

## SUMMARY:

### •Key Concepts:

- Heat dissipation and material limitations in hypersonic speeds
- Aerodynamic instability and control challenges
- Scramjet engines as propulsion systems

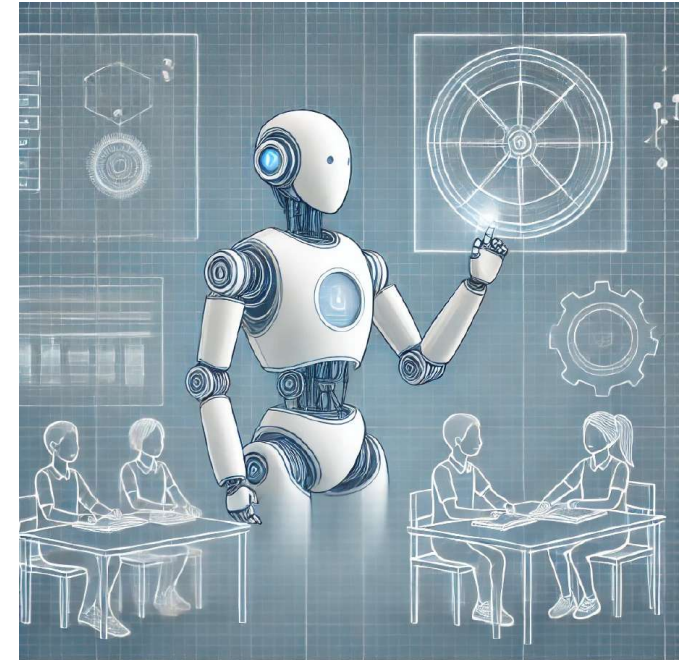
### •Applications:

- Development of next-gen military aircraft and spaceplanes
- Faster transport options for intercontinental travel

### •Takeaways:

- Innovation in thermal protection systems is pivotal.
- Research in predictive modeling for aerodynamic behavior is ongoing.

Now summarize the following article or concept while maintaining this level of structure and clarity: [Input material].



# Iterative Prompting (Progressive Refinement)

## WHAT IS IT?

Technique for gradual improvement of responses

Step-by-step refinement process

Each iteration improves on the previous one

Similar to the academic review process

## WHY DOES IT WORK?

Allows for progressive adjustments

Maintains context across iterations

Facilitates specific deep dives

Improves final quality

## ACADEMIC APPLICATIONS:

Development of new courses

Academic writing

Research design

Material enhancement

## ITERATIVE PROMPTING

### Basic Structure:

#### 1. Broad Initial Prompt

Begin with a general request to establish a starting point.

#### 2. Response Analysis

Identify strengths and areas for improvement.

#### 3. Specific Refinement

Focus the prompt on weak points or details.

#### 4. Deep Dive into Key Areas

Final iteration explores critical components.

# Example of iterative prompting

## **ACADEMIC USE CASE:**

"I need to create a structured teaching module for an undergraduate course on Orbital Mechanics, focusing on Hohmann transfer orbits. Use Latex equations in your response."

## **ITERATIVE PROMPTS:**

### **ITERATION 1 - INITIAL STRUCTURE:**

"Propose an outline for a teaching module on Hohmann transfer orbits. Include:

- Key topics to cover,
- A breakdown of sessions,
- Time allocation for each session."

*[Receive initial outline]*

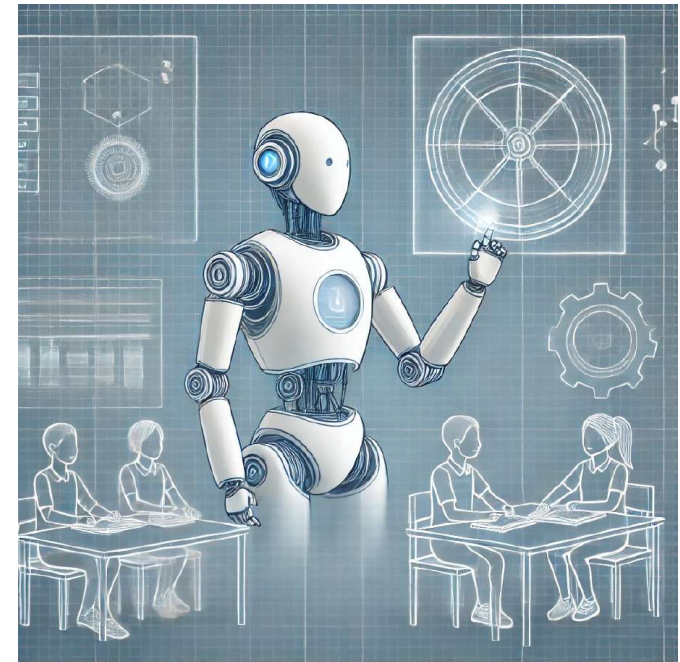
### **ITERATION 2 - ENHANCING DETAIL:**

"Good outline! Now focus on [example Session 2: 'Mathematical Derivation of the Hohmann Transfer'.]

Expand with:

- Specific equations to derive,
- Examples or problems to solve,
- Supporting diagrams to include in teaching materials."

*[Receive detailed plan for Session 2]*



# Example of iterative prompting

## **ITERATION 3 - ACTIVE LEARNING STRATEGIES:**

"Great detail in Session 2. Now refine it further by integrating active learning:

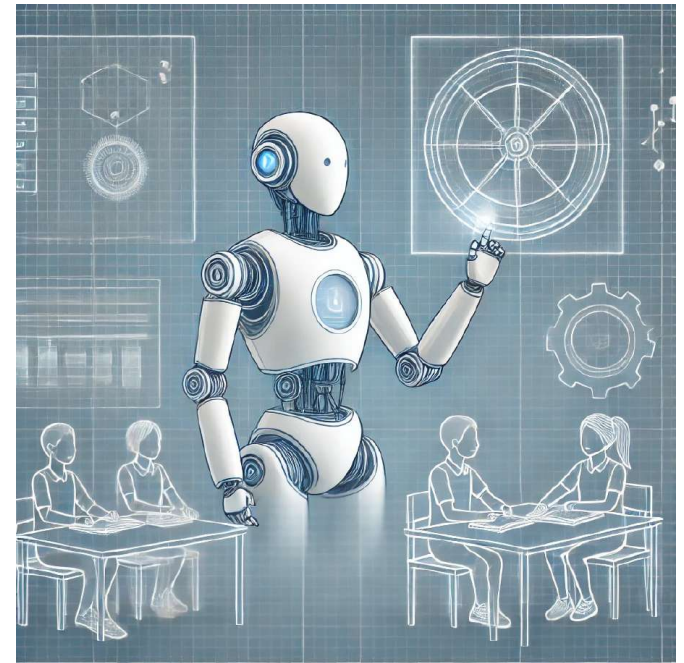
- Suggest in-class activities (e.g., group problem-solving, simulations),
- Homework assignments,
- Assessment questions related to the derivations."

*[Receive list of active learning strategies]*

## **ITERATION 4 - FINAL TOUCHES:**

"Add specific tools or software recommendations to assist in teaching this module. For example, use MATLAB or Python scripts to simulate a Hohmann transfer. Give me the scripts."

*[Receive finalized plan with software tools]*





# Meta-Prompting

## (Using AI as a Prompt Consultant)

### WHAT IS IT?

A technique where AI helps design effective prompts.

Leverages the LLM's understanding of itself.

Optimizes human-AI communication.

Iteratively improves instructions.

### WHY DOES IT WORK?

AI understands its own capabilities.

Can suggest optimal structures.

Identifies key missing elements.

Helps avoid common pitfalls.

Unexpected insight.

### ACADEMIC APPLICATIONS:

Designing activities with AI.

Improving instructional clarity.

Optimizing queries for research.

Developing course materials.

### Basic Structure:

**1. Broad Initial Task:** Provide a general description of what you want to achieve, even if it's vague.

**2. Ask the AI to Lead the Process:** Use a meta-prompt like: "Ask me all the questions you consider necessary to design the best possible prompt for this task. Use those answers to craft a detailed and effective prompt."

**3. AI-Asks-You Questions:** Let the AI generate its own clarifying questions without interference.

**4. AI-Crafts the Final Prompt:** After answering the questions, instruct the AI to generate the complete and actionable prompt.

# Example of meta-Prompt

I want to design an assignment for my students on [topic: solving linear differential equations] with the following features:

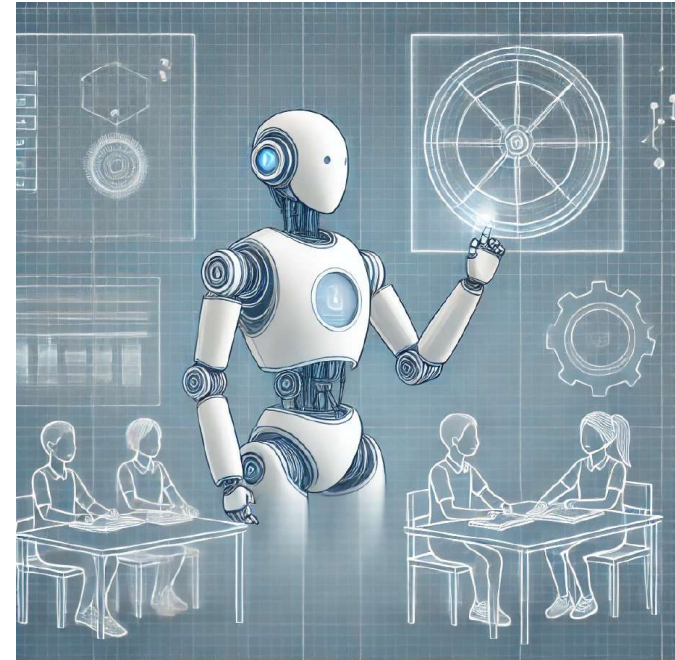
1. Engages students in problem-solving.
2. Includes real-world applications.
3. Aligns with undergraduate-level understanding.

I don't have much experience creating prompts for AI. Could you ask me all the questions you consider necessary to help me define a clear and effective prompt? Be very exhaustive and ask me everything you can think about.

Once we have all the answers, generate the final prompt that I can use for the AI.

**Some questions can surprise us.**

**After you respond, the LLMs tend to answer the prompt, not give it. Ask it again to give you the prompt since it can be later reused.**



# System Prompt (general instructions)

## SYSTEM PROMPT:

"You are an expert academic assistant, specialized in working with professors and researchers in engineering. Your responses should be:

Technical but clear  
Academically rigorous  
Practical and applicable  
Ethically conscious

## You excel at:

Course design and planning  
Research methodology  
Technical documentation  
Academic writing  
Educational innovation

## When interacting:

Always ask for clarification if the academic context is unclear  
Consider pedagogical implications  
Provide evidence-based suggestions  
Maintain academic standards  
Encourage critical thinking

## Avoid:

Oversimplified answers  
Non-academic sources  
Unsubstantiated claims  
Biased perspectives

## Format responses using:

Clear structure  
Academic citations when relevant  
Examples from engineering context  
Technical terminology with explanations"



## Exercise in advanced prompting

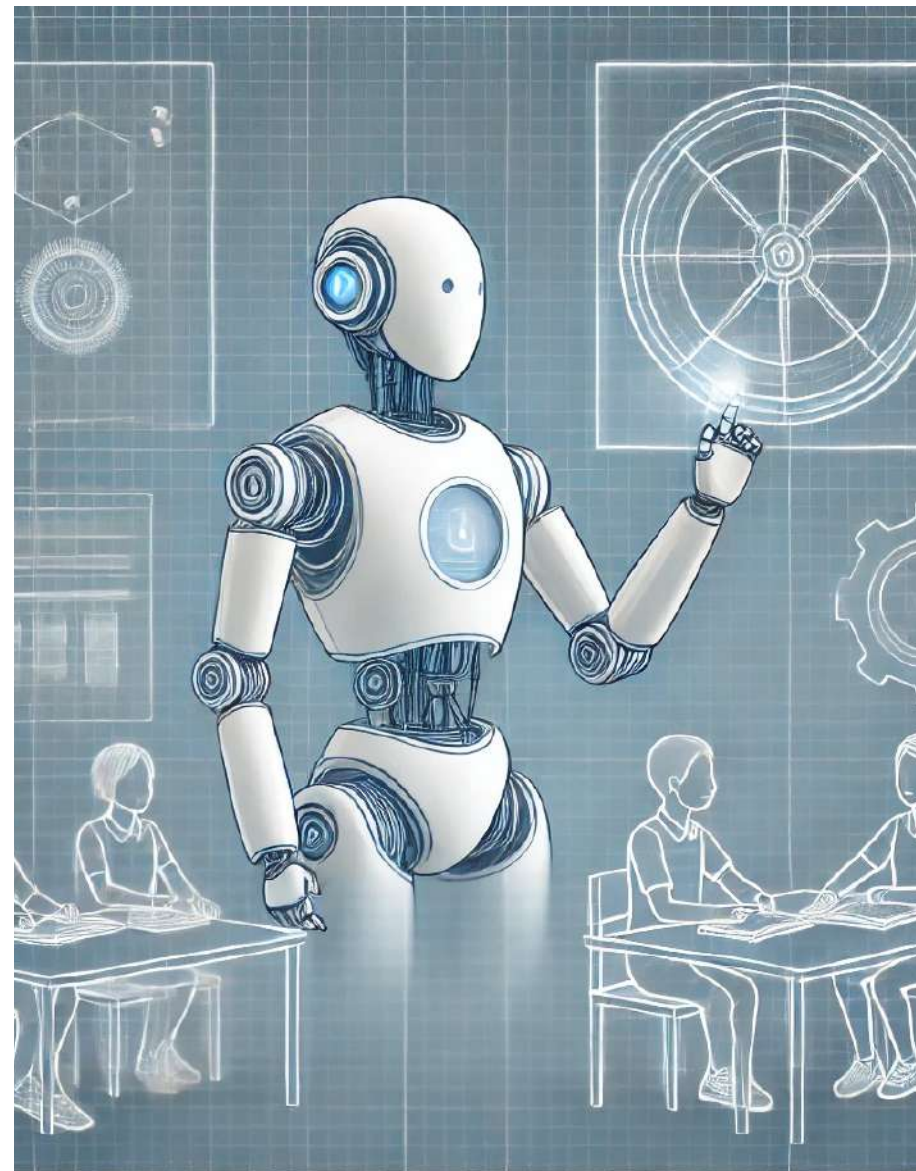
---

The best way to master these techniques is by applying them to real challenges. Choose a challenge from your academic activity:

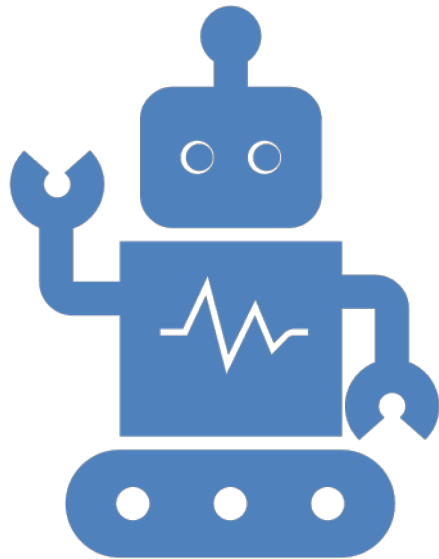
- **Teaching:** Design an activity, create materials, plan evaluations.
- **Research:** Develop methodologies, analyze data, write an article abstract.
- **Administration:** Organize events, write an email, prepare documentation.

Select the technique that best suits your needs and design an effective prompt (or have the AI design it for you!).

**Share your experience:** what technique you chose, why, and what results you achieved.







# Outline

1. A personal journey into AI
2. How LLMs work
3. Basic Prompt Engineering
4. Updated LLM Ecosystem (Dec 2024)
5. Limitations of LLMs
6. Advanced Prompt Engineering
7. **Generating documents**
8. Generating visual presentations
9. Generating code
10. Other use cases and tools
11. Closing and discussion

# Writing Papers with Gen AI

## Why Use Gen AI in Writing Academic Texts?



- Gen AI offers assistance from the beginning in structuring and developing documents.
- Ability to generate content, ideas, and provide explanations. **Fighting the writer's block.**
- A complementary tool for efficiency.
- The start: generating an index.

## Example Prompt

"As an expert in space engineering, generate a detailed index for a paper on space debris, including theoretical sections and practical applications, in LaTeX and providing the generated code."

## Selection and Customization of Templates

- The second step: generating a document template.
- Gen AI's help in choosing and adapting LaTeX templates.
- Tips for customizing and enriching the document.

### Example Prompt

"Recommend a visually appealing LaTeX template for an engineering project report, and explain how to customize it."

# Iteration and Content Enrichment

## Document Development and Improvement

- Next steps: Using gen AI to draft and develop arguments section by section.
- The importance of reviewing and customizing gen AI's contributions.
- Iterating when not satisfied with the results. Even changing the approach.

### Example Prompt

"Act as an aerospace engineering expert. Provide an initial draft for a section on the environmental impacts of space propulsion in an aerospace engineering paper."

### Example Prompt

"Improve section X, specifically focusing on the impact generated by a launch, providing more details."



## Deepening and Diversifying Content

- Active search for external sources to avoid shallowness in the topics covered.
- Using gen AI to summarize and adapt these external contents to the document, improving the depth and richness of the work.
- Essential: All material from external sources used must be duly cited to maintain academic integrity and respect copyright.

### Example Prompt

"Based on the following excerpt from a study on chemical propulsion, summarize the key points that could enrich my work on aerospace engineering. Be sure to include a proper citation to the study in the required format."

# Verification and References

## Ensuring Accuracy and Reliability

- Independent verification of the information provided by gen AI: a crucial step.
- Use of primary sources and review of relevant literature: Google, Google Scholar, others.
- Emphasis in particular on those things we do not control: better not to trust too much! It's better to omit and admit than to lie.
- Leveraging LaTeX capabilities to generate appropriate formats (bibtex).

## Example Prompt

"Suggest some recent references on artificial intelligence in aerospace engineering. Generate the corresponding bibtex file for inclusion in LaTeX."

**Note:** Verify all generated references!



# Integrating Formulas and Graphics

## Enhancing Presentation

- Since LaTeX can be considered code, gen AI is capable of assisting in producing quality texts in this format by integrating complex elements.
- Gen AI's assistance in including visual elements in LaTeX.
- Generating code for figures, tables, and equations.
- Resolving LaTeX errors.

### Example Prompt

"Generate LaTeX code for a table showing performance data of different types of space thrusters."

### Example Prompt

"Show an example of LaTeX code for a customized, visually appealing cover for an aerospace engineering report."

# Collaboration and Review in Overleaf

Overleaf is a collaborative online editing platform for LaTeX that can be used for free (up to a certain document size).

## Advantages:

- Real-Time Collaboration: Overleaf allows multiple users to work on the same document simultaneously, facilitating collaboration on group projects and document review.
- Ease of Use: Overleaf features an intuitive interface and editing options that make LaTeX more accessible.

## Optimizing Teamwork

- Gen AI as a facilitator in project collaboration on Overleaf.
- Strategies for joint document review and editing.

## Example Prompt

"Suggest some strategies for effective collaboration in writing a group report in Overleaf."





# Tips for Effective Academic Writing

## Improving Writing Skills

- Gen AI as a resource to strengthen academic writing.
- Strategies for solid arguments and content review.
- Summaries, conclusions, text unification, rewriting.

### Example Prompt

"Provide tips for improving coherence and fluency in scientific article writing."

### Example Prompt

"Apply the previously provided tips to this text block and explain the changes you've made."

### Example Prompt

"Merge the second and third paragraphs so that the text flows better, and also generate a summary of the text."

# Balancing Assistance and Personal Knowledge



## Complement, Not Substitute!

- The importance of ethical and responsible use of gen AI.
- Gen AI as a complement to personal effort and learning.
- It is impossible to detect with certainty that a text has been produced by an LLM, but it is possible to detect a poorly done text or with false errors/references, plagued with pomposity and exaggerations!



# Managing gen AI's Limitations

## Practical Tips in the Face of Challenges and Limitations:

- **Token Limit:** Divide extensive topics into manageable segments to maintain coherence.
- **Hallucinations:** Always verify the information provided with external sources.
- **Updated Knowledge and Internet Access:** Supplement with searches in recent sources to obtain updated data.
- **Biases and Fairness:** Be critical and aware of potential biases in the responses.
- **Prompt Quality Dependence:** Formulate clear and specific prompts to obtain more relevant and coherent responses.
- **Shallowness:** Supplement the creative process with external sources to enrich content and ensure depth.
- **Inappropriate Language:** Carefully review the generated text and adjust the tone by formulating appropriate prompts.
- **Security and Privacy:** Avoid sharing sensitive information and stay informed about privacy and security policies.



# Text to Documents (Automatic Generation)

---

## WHY IS IT RELEVANT?

- Speeds up the creation of teaching materials.
- Standardizes documentation.
- Adapts content effectively.

## KEY USE CASES

- Laboratory guides.
- Teaching plans.
- Evaluation rubrics.
- Technical documentation.

## LIMITATIONS

- Requires expert review.
- Limited personalization.
- Consistency challenges in long documents.

## TOOLS

- ChatGPT 4.0
- Microsoft Copilot Pro





# Practical exercise – choose your task

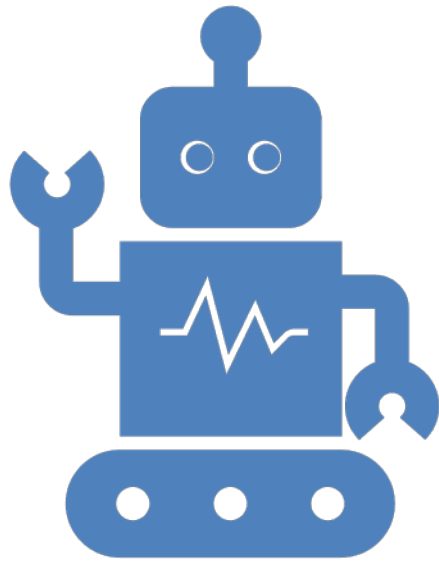
## Exercise 1: Course Guide Creation

- **Scenario:** Generate a teaching guide for a university course.
- **Base Prompt:**  
\*"As a course coordinator, create a teaching guide for [SUBJECT]. Include: Objectives and competencies, Teaching methodology, Evaluation system, Recommended bibliography  
Use a formal tone, ensuring compliance with academic standards."\*
- **Steps:**
  1. Adapt the prompt to a specific course, maybe add downloaded materials for quality
  2. Generate the document and refine iteratively using AI.

## Exercise 2: Summary Report Creation

- **Scenario:** Create a summary report on a research topic or case study.
- **Base Prompt:**  
\*"Write a summary report on [TOPIC]. Include: Context and background, Key findings or concepts, Practical implications  
Format: Concise and formal."\*
- **Steps:**
  1. Select a topic relevant to your field (e.g., AI in engineering or a recent academic paper).
  2. Research briefly (or use downloaded materials), then use AI to generate and organize the report.

Choose one task and apply AI to create professional, high-quality documents!



# Outline

1. A personal journey into AI
2. How LLMs work
3. Basic Prompt Engineering
4. Updated LLM Ecosystem (Dec 2024)
5. Limitations of LLMs
6. Advanced Prompt Engineering
7. Generating documents
8. **Generating visual presentations**
9. Generating code
10. Other use cases and tools
11. Closing and discussion

# Text to Image

## WHY IS IT RELEVANT?

- Rapid creation of educational visual materials.
- Illustration of complex technical concepts.
- Customization of teaching resources.

## KEY USE CASES

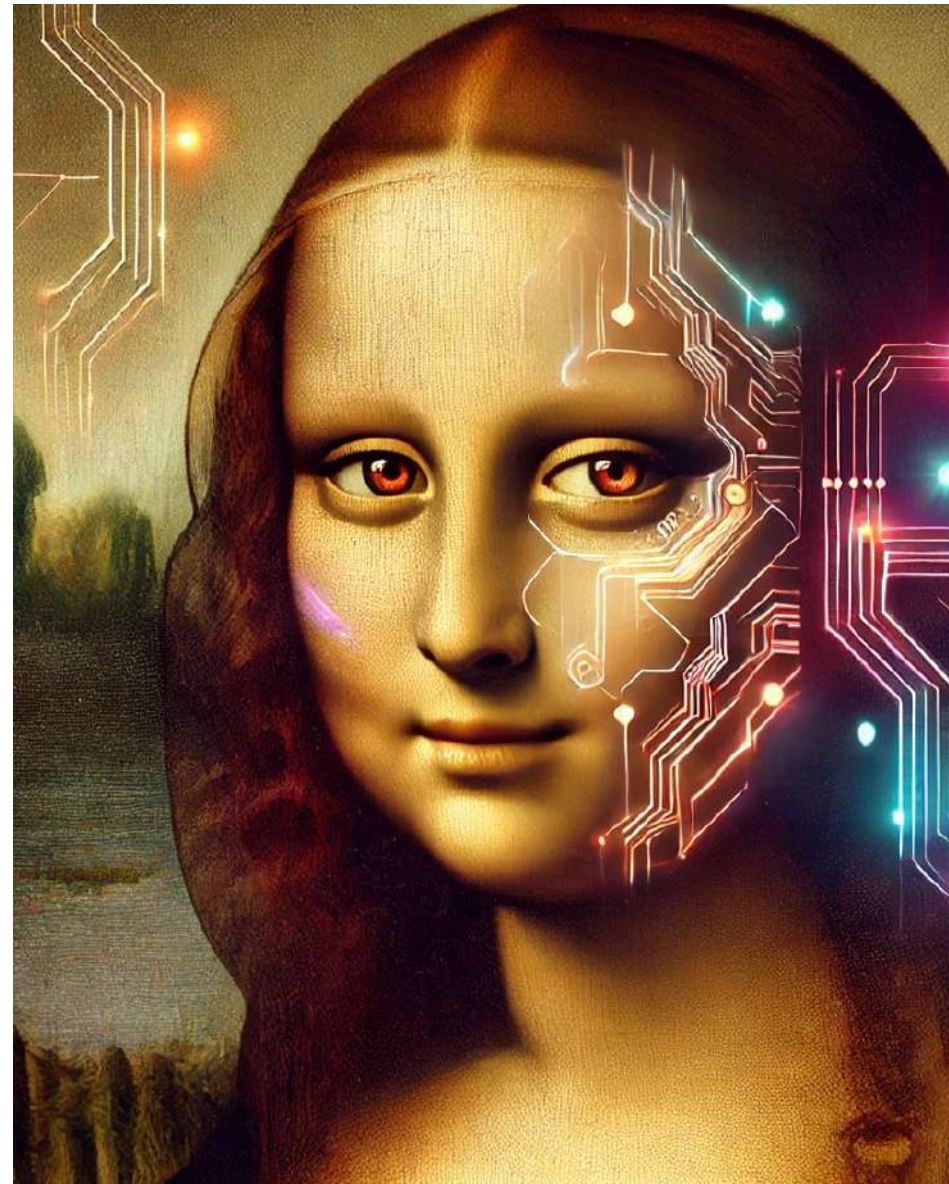
- Technical diagrams for notes... but not always reliable.
- Illustrations of engineering processes.
- Conceptual visualizations.
- Materials for presentations/posters.

## LIMITATIONS

- Low accuracy in technical details.
- Generated text is often unreliable.
- Lack of consistency between images.
- Unclear usage rights and attribution issues.

## TOOLS

- DALL-E 3
- Imagen 3
- Ideogram
- Recraft
- MidJourney
- Stable Diffusion



# Image generation tools

## DALL-E 3

- **Access:** ChatGPT
- **Strengths:**
  - Excellent understanding of prompts.
  - Consistent results.
  - Ideal for corporate-style images.
- **Limitation:** Not the best results overall.

## IMAGEN 3 (Google)

- **Access:** Gemini
- **Cost:** Free
- **Strengths:**
  - Simple interface.
  - Good technical results.
  - No usage limitations.
- **Limitation:** Does not allow human images in the free plan.



## IDEOGRAM.AI

- **Access:** Direct web access
- **Cost:** Free credits upon registration
- **Strengths:**
  - Excellent with text and logos
  - Highly professional style
  - Advanced style control
- **Limitation:** Credit system, very slow



Recraft

## RECRAFT.AI

- **Access:** Direct web access
- **Cost:** Free plan available
- **Strengths:**
  - Currently top-rated for generation
  - SVG generation
  - Icons and UI elements
  - Fast variations generation

The logo for DALL-E, featuring the text "© OpenAI DALL·E" in a white, sans-serif font on a black background.

© OpenAI  
DALL·E



# Example

## Example: "The Augmented Engineer"

### PROMPT:

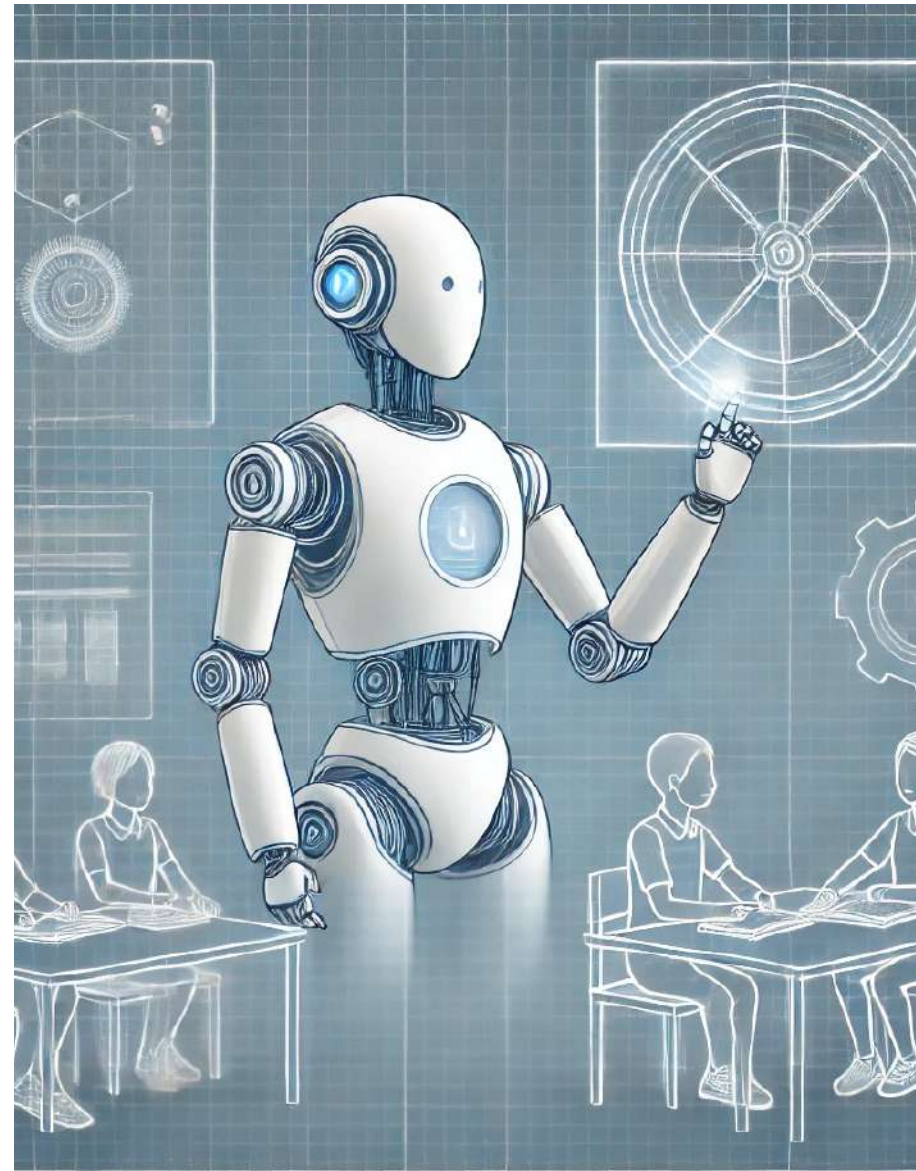
"A futuristic engineer in their laboratory, surrounded by multiple floating interactive holograms. The holograms display glowing mathematical formulas, 3D models of structures, and data charts. The engineer is dressed casually yet professionally and gestures as they interact with the visualizations. Modern artistic style, vibrant yet professional colors, technological but welcoming atmosphere."

### Other Ideas:

- "The Living Library" (for document-to-text transformation)
- "The Data Whisperer" (for data analysis)
- "The Digital Architect" (for image-to-video transformation)

### Suggestions:

- Use **metaprompting** if necessary.
- Compare outputs across **ChatGPT 4.0**, **Recraft**, and **Ideogram** for the best results.



# Introduction to Preparing Presentations with Beamer

## Transforming Documents into Visual Presentations

- Beamer: A LaTeX class specialized for creating slides.
- Gen AI can assist in converting document content into attractive presentations.
- Gen AI can serve as a tool for suggesting titles and key content.
- Gen AI can assist with complex formatting aspects (see the cover of this document).

This document is an example generated from the practice report. The reverse process is also possible (generating text from a presentation) but will typically require more effort (adding more detail).

## Optimizing Slide Appearance

- First step: Gen AI suggestions on template, color schemes, and design.
- Tips for effective text and visual element arrangement.

### Example Prompt

"Based on the document I provide, suggest a format for a Beamer presentation in LaTeX, including an attractive title, color scheme, and cover. Pass me the code for that presentation."

## Effectively Creating Slides with Gen AI

- Using Gen AI to transform text from sections into slides, step by step.
- Focus on key ideas and clear, concise presentation. Iterate until satisfied/retouched.

## Highlighting the Essentials on Each Slide

- Identify and highlight key points from each section.
- Avoid overloading slides with excessive text.

## Example Prompt

"Begin generating the document with the introduction section. I pass you the LaTeX of that section. There are two figures, it would be great if somehow one of them could be integrated into the cover. Please return the revised LaTeX code of the cover as well as the proposal for the introduction section for the presentation. Be concise but do not omit relevant information."



## Enriching Slides with Graphic Elements

- Use of images and graphics to reinforce concepts.
- Identifying opportunities to include graphics and tables.
- Enhancing audience comprehension and retention.

### Example Prompt

"Generate LaTeX code to include a graph illustrating elliptical orbits in a Beamer slide."

# Review and Final Adjustments

## Ensuring Quality and Effectiveness

- **Consistency and Coherence:**  
Review of style, format, and narrative.
- **Clarity and Simplicity:** Simplification of slides to highlight key messages.
- **Visual and Aesthetic Impact:**  
Improvements in arrangement, image quality, and professional appearance.



## Example Request to Gen AI

"Review the coherence and visual impact of my computational fluid dynamics presentation slides. Suggest improvements if you see any possible."

# Tips for an Effective Presentation

## Beyond the Slides

- Guidance on communication and public speaking skills.
- Techniques to engage and maintain audience attention.
- Strategies for managing Q&A sessions effectively.

### Example Prompt

"Provide tips for successfully managing a Q&A session after a technical presentation."

### Example Prompt

"Act as a professor very critical with his students. Based on the presentation I provide, generate five questions and propose answers to them."



# Generating Powerpoint Slides

## KEY ELEMENTS

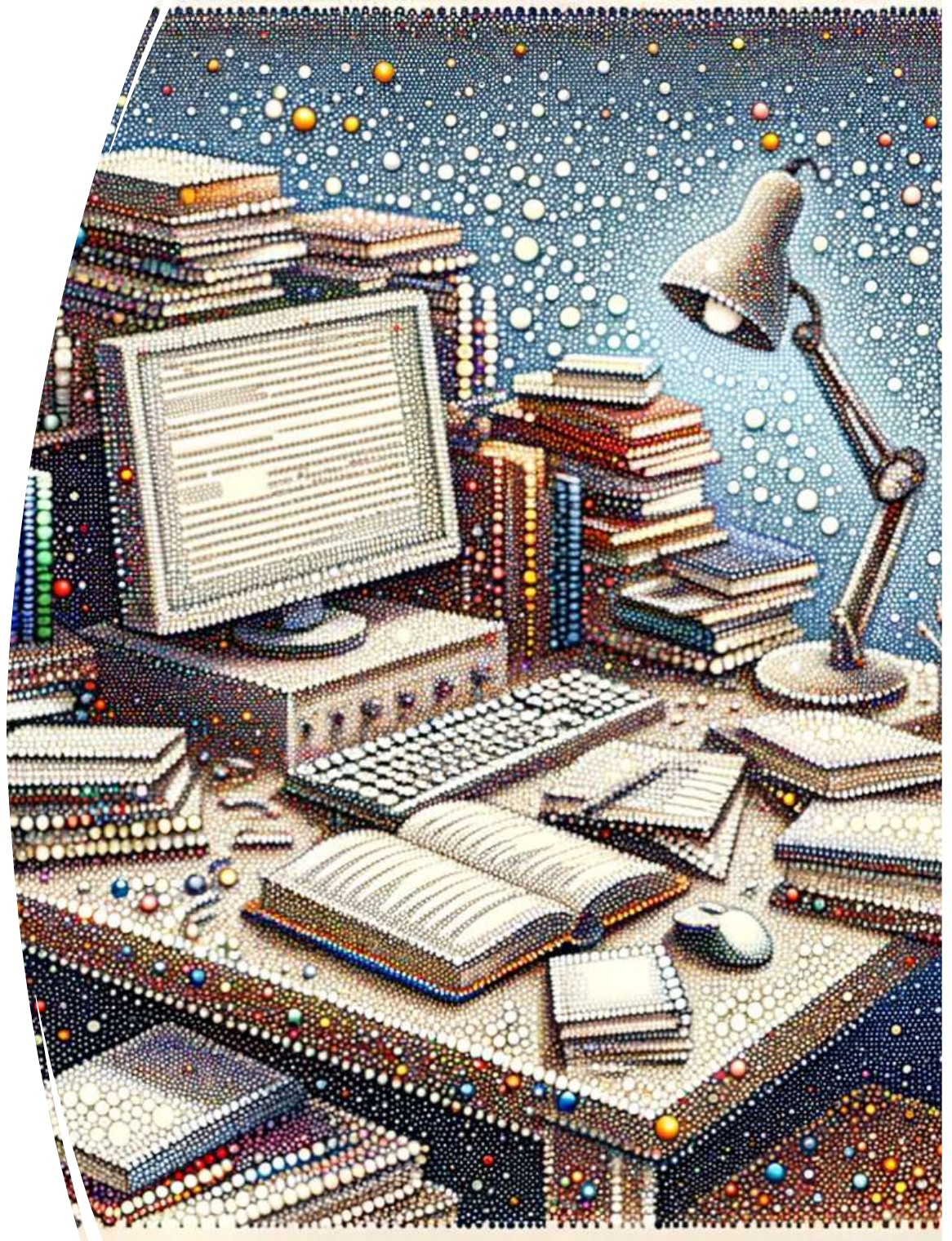
- Clear structure
- Coherent narrative
- Effective visualization
- Impactful points

## AI TOOLS

- LLMs for structure
- Image generation for visuals
- PPT designers for layout

## BEST PRACTICES

- One idea per slide
- Visualized data
- Concise text
- Consistent design
- Start from a technical document





# Workflows to Generate Presentations from a Document



Generate outline



Go slide by slide obtaining the  
different points



Add visual elements



Beamer: better for equations



Powerpoint: better for visuals

# Practical Exercises – Choose Your Task

## Exercise 1: Research Results Presentation

- **Scenario:** Create a 20-minute presentation on experimental results related to **XX**, including:
  - **Methodology**
  - **Data Analysis**
  - **Visualizations**
- **Objective:** Communicate research findings clearly and effectively.

## Exercise 2: Mini Teaching Presentation

- **Scenario:** Create a 10-minute presentation to teach a concept in your area, including:
  - **Engaging Introduction**
  - **Step-by-Step Explanation**
  - **Practical Example**
  - **Quick Evaluation**

# Practical Exercises – Choose Your Task

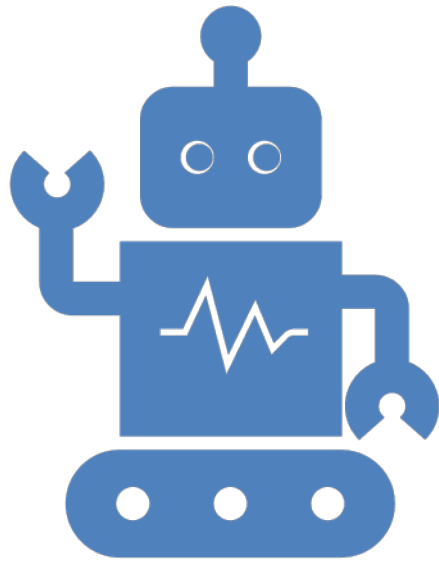
## Steps for Both Exercises

- 1. Structure Content:** Use AI to organize sections and generate slide outlines.
- 2. Generate Key Slides:** Prompt AI to draft slides with concise content for Beamer or PowerPoint.
- 3. Add Visual Elements:** Use AI tools (e.g., ChatGPT for charts or DALL-E for visuals).
- 4. Review and Refine:** Polish the layout and ensure clarity.

## Tools:

- **Beamer (Overleaf) or PowerPoint (Designer/)**
- **AI Prompts for Slide Content**

Choose your task and follow the steps to create a professional presentation!



# Outline

1. A personal journey into AI
2. How LLMs work
3. Basic Prompt Engineering
4. Updated LLM Ecosystem (Dec 2024)
5. Limitations of LLMs
6. Advanced Prompt Engineering
7. Generating documents
8. Generating visual presentations
- 9. Generating code**
10. Other use cases and tools
11. Closing and discussion



# Using Gen AI in Programming

## Enormous Potential for Code Generation

- Tool for automating tasks and assisting in debugging.
- Useful in education for improving understanding and application of concepts.

## Generating Codes with Gen AI in MATLAB

- Generation of scripts and mathematical functions.
- Creation of advanced visualizations and use of specific functions.
- Providing detailed explanations of generated codes.

Not without dangers...

# Data Visualization Example in MATLAB

## Objective:

- Visualize functions in MATLAB with attractive visualization options and sufficient size (common mistake).

## Prompt:

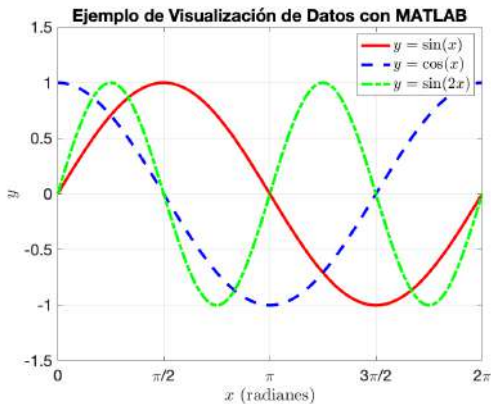
Hello, ChatGPT. I'm working on a MATLAB project and need your help to generate a specific graph. Specifically, I need a MATLAB graph that shows three different functions:  $y = \sin(x)$ ,  $y = \cos(x)$ , and  $y = \sin(2x)$ , in the range from 0 to  $2\pi$  radians. For each function, use different line styles and colors. Please include a legend with the labels of each function written in LaTeX, and label the axes also in LaTeX. The graph's title should be "Data Visualization Example with MATLAB". Ensure the font sizes are large enough to be legible even when the image is reduced to half its original size. I would also like you to add a grid to make the graph easier to read. Finally, please include the code to save the graph as a PNG file.

# MATLAB Code for Data Visualization

```
% Function definitions
x = linspace(0, 2*pi, 1000);
y1 = sin(x);
y2 = cos(x);
y3 = sin(2*x);
figure; % Creating the figure
% Plotting the functions
plot(x, y1, 'r-', 'LineWidth', 3);
hold on;
plot(x, y2, 'b--', 'LineWidth', 3);
plot(x, y3, 'g-.', 'LineWidth', 3);
% Adding legend with LaTeX code
legend({'$y = \sin(x)$', '$y = \cos(x)$', '$y = \sin(2x)$'}, 'Interpreter', 'latex', 'FontSize', 16);
% Adding labels with LaTeX code
xlabel('$x$ (radians)', 'Interpreter', 'latex', 'FontSize', 18);
ylabel('$y$', 'Interpreter', 'latex', 'FontSize', 18);
% Graph title
title('Data Visualization Example with MATLAB', 'FontSize', 20);
% Setting axis limits
xlim([0 2*pi]), ylim([-1.5 1.5]);
% Adding grid
grid on;
% Configuring axis ticks
xticks(0:pi/2:2*pi);
xticklabels({'0', '\pi/2', '\pi', '3\pi/2', '2\pi'});
yticks(-1.5:0.5:1.5);
% Additional configuration to improve appearance
set(gca, 'FontSize', 16); % Axis font size
% Saving the figure
saveas(gcf, 'data_visualization_example.png');
```



# Generated Visualization in MATLAB



Better presentation than in 90% of the students' academic works corrected by the professor, with clarifying comments in the code!





# Example of Using Unknown Functions

## Objective:

- Sometimes one may have a need and not know which function to use (important to be very specific and maybe you'll hit the nail on the head!).

## Prompt:

Hello, ChatGPT. I need your help to create a visualization in MATLAB. I have a set of 10 points randomly distributed in a 100x100 space. I would like to generate regions in this space so that each region is closer to one of these points than to the others. For this, draw lines that separate these regions. Mark the points with red dots and the lines with blue. Also, include labels for the axes and a descriptive title. Please make sure the graph is clear and easy to understand. Finally, generate a PNG with the result. Thank you for your help.

# Resulting MATLAB Code

```
% Generate random points
points = 100 * rand(10, 2);

% Calculate Voronoi diagram
[v, c] = voronoin(points);

% Create figure
figure;
hold on;

% Draw Voronoi diagram lines
[vx, vy] = voronoi(points(:,1), points(:,2));
plot(vx, vy, 'b-', 'LineWidth', 1.5); % Draw lines

% Draw points
plot(points(:,1), points(:,2), 'r.', 'MarkerSize', 20); % Draw points

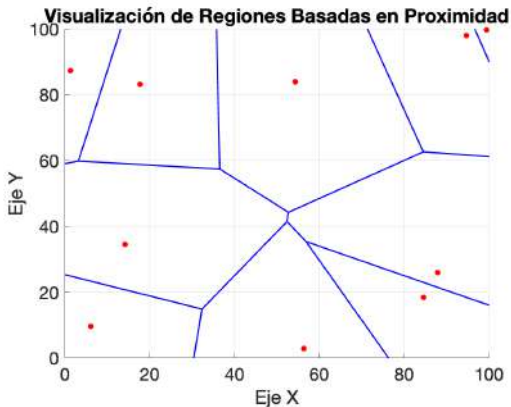
% Customize the graph
title('Visualization of Proximity-Based Regions', 'FontSize', 18);
xlabel('X Axis', 'FontSize', 18);
ylabel('Y Axis', 'FontSize', 18);
set(gca, 'FontSize', 18); % Axis font size
xlim([0 100]);
ylim([0 100]);
grid on;

% Save the figure as PNG
saveas(gcf, 'voronoi_diagram.png');

hold off;
```



# Generated Visualization in MATLAB



The student discovers the concept of the Voronoi diagram thanks to this code.



# Further Exploration...

What is a Voronoi diagram?

**Prompt:**

define, for a 4th-year aerospace engineering student, what is a Voronoi diagram, thanks

**Answer:**

A Voronoi diagram is a mathematical tool used to divide a space into regions based on proximity to a set of specific points. Imagine you have a group of points scattered in a space, like stars in the sky. The Voronoi diagram allows you to create boundaries around each point so that any point within a region is closer to that point than to any other in the set. These regions are called "Voronoi cells".

In the context of aerospace engineering, Voronoi diagrams are used to solve problems related to resource distribution or route planning. For example, if you are planning a space mission and have a network of ground stations, a Voronoi diagram would help you determine which of these stations is closest to a specific location in space, which can be crucial for spacecraft communication or navigation.

# Python Automation Example

## Objective:

- Automate document generation using Python, learning about its use for document manipulation.

## Prompt:

Hi chatgpt. Do you know any way to automate the generation of a Word document? I have a base document, where it is necessary to write the name and ID of several people in various places. I separately have the list of people and IDs. I would like a separate document to be generated for each one with this information replaced. Provide a Python script that performs this task easily and quickly.



# Python Code for Document Automation

```
from docx import Document

def fill_template(template_path, personal_data, output_dir):
    for person in personal_data:
        # Load the template
        doc = Document(template_path)
        # Replace placeholders
        for paragraph in doc.paragraphs:
            if '{name}' in paragraph.text:
                paragraph.text = paragraph.text.replace('{name}', person['name'])
            if '{ID}' in paragraph.text:
                paragraph.text = paragraph.text.replace('{ID}', person['ID'])
        # Save the document
        output_path = f"{output_dir}/{person['name']}_document.docx"
        doc.save(output_path)

# Example data and template path
template_path = 'path/to/your/template.docx'
personal_data = [{'name': 'John Doe', 'ID': '12345678A'},
                 {'name': 'Anna Smith', 'ID': '23456789B'}]
output_dir = 'path/to/the/generated/documents'

fill_template(template_path, personal_data, output_dir)
```



# Python Automation Result

- This Python script demonstrates an efficient way to automate the generation of customized Word documents.
- Uses the “python-docx” library, which is easy to install (you can even ask ChatGPT how to do it).
- The result is fast, simple, and highly adaptable to various document automation needs.

# Risks and Considerations in Code Generation

- Codes that are syntactically correct but may contain logical errors.
- Risk of dependency and limitation in skill development.
- Importance of originality and personal understanding in educational contexts: if we don't learn anything, what are we here for?
- The proposed code may not be optimal or have implicit assumptions we are unaware of (e.g., discretizations for solving differential equations).
- "Sometimes there's no way": ChatGPT keeps producing garbage. Rethink the question, formulate it in different ways starting from scratch again... or admit that you'll have to do the work yourself!

# Coding Help

## BASED ON YOUR NEED:

### 1. I WANT TO UNDERSTAND

#### ChatGPT:

1. Explains step by step.
2. Answers questions.
3. Provides simple examples.

### 2. I WANT TO CREATE/MODIFY

#### GitHub Copilot / Codium:

1. Completes code as you type.
2. Suggests solutions.
3. Integrated into your editor.

### 3. I WANT TO EXPERIMENT

#### Google Colab + Gemini:

1. Ready-to-use environment.
2. No installations required.
3. Safe trial and error.

### 4. I NEED SOMETHING COMPLEX

#### Claude:

1. Handles longer code.
2. Provides detailed explanations.
3. Better context management.



# Help coding

## HOW TO ASK FOR HELP WITH CODE

### STRUCTURE YOUR REQUEST:

1. Explain **WHAT** you want to do.
2. Describe the **CONTEXT** (files, data, etc.).
3. Mention your **skill level** ("I'm a beginner...").
4. Request **explanations** if needed.

### GOOD EXAMPLES:

- ✓ "I need to read an Excel file and send emails. I'm a beginner, please explain step by step."
- ✓ "Here's my code [code snippet], it gives an error. What does it mean?"
- ✓ "How can I adapt this so it also does...?"

### EXAMPLES TO IMPROVE:

- X "Give me code for Excel."
- X "It doesn't work, help."
- X "Make this more efficient."



## Advice for non-programmers

### **1. DIVIDE AND CONQUER**

- Break the problem into parts.
- Ask questions for each part.
- Integrate step by step.

### **2. ITERATION IS YOUR FRIEND**

- Start simple.
- Test to ensure it works.
- Add functionalities incrementally.

### **3. LEARN BY ASKING**

- "What does this line do?"
- "Why did you use X instead of Y?"
- "Is there a simpler way?"

### **4. SAVE AND DOCUMENT**

- Comment on successes.
- Save what works.
- Note common errors.

# Risks and limitations

## **RISKS AND LIMITATIONS**

### **1. GENERATED CODE**

- May contain errors.
- Requires verification.
- Needs adaptation.

### **2. LEARNING**

- Balance copying with understanding.
- Ask "why" for every solution.
- Customize to fit your needs.

### **3. PRACTICE**

- Start with small examples.
- Modify existing code.
- Learn from mistakes.

### **4. THE PROCESS**

- No solution is perfect.
- Iteration is part of the process.
- Ask for help when needed.

# Prompting example

## Basic Prompt (BAD):

"Code to solve a differential equation."

## Improved Prompt (GOOD):

"Generate Python code to solve the differential equation  $dy/dx + 2y = x^2$  using the Euler method.

### Requirements:

Use **NumPy** for calculations.

Include visualization of the solution using **Matplotlib**.

Add **explanatory comments** throughout the code.

Incorporate **result verification** against an analytical solution.

Tailor the code for **second-year students** (beginner-friendly).

Ensure the code is **modular** and adheres to **PEP 8** standards."

Try the code on <https://colab.research.google.com/>

# Exercise

## Practical Case: Virtual Laboratory

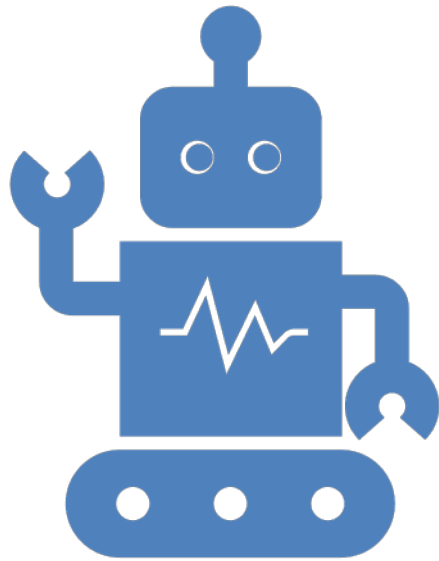
### Scenario:

Develop code focused on:

1. **Simulating a Physical System:** Use models like pendulums, electrical circuits, or heat transfer, or your favorite system.
2. **Experimental Data Analysis:** Generate or provide experimental datasets for analysis.
3. **Result Visualization:** Create clear and meaningful plots (e.g., time series, histograms, 3D models).
4. **Automated Reporting:** Automatically generate a report summarizing the findings, including key metrics and visualizations.

### Tools:

- **Programming Languages:** Python or MATLAB.
- **Interactive Environments:** Jupyter Notebook or Google Colab.
- **Scientific Libraries:** NumPy, SciPy, pandas, or MATLAB toolkits.
- **Graphing Tools:** Matplotlib, Seaborn, or MATLAB's visualization tools.



# Outline

1. A personal journey into AI
2. How LLMs work
3. Basic Prompt Engineering
4. Updated LLM Ecosystem (Dec 2024)
5. Limitations of LLMs
6. Advanced Prompt Engineering
7. Generating documents
8. Generating visual presentations
9. Generating code
- 10. Other use cases and tools**
11. Closing and discussion





# NotebookLM

What is NotebookLM?

- AI specialized in source analysis
- Accurate and verifiable citations
- Specialized contextual memory
- Free (with a Google account)

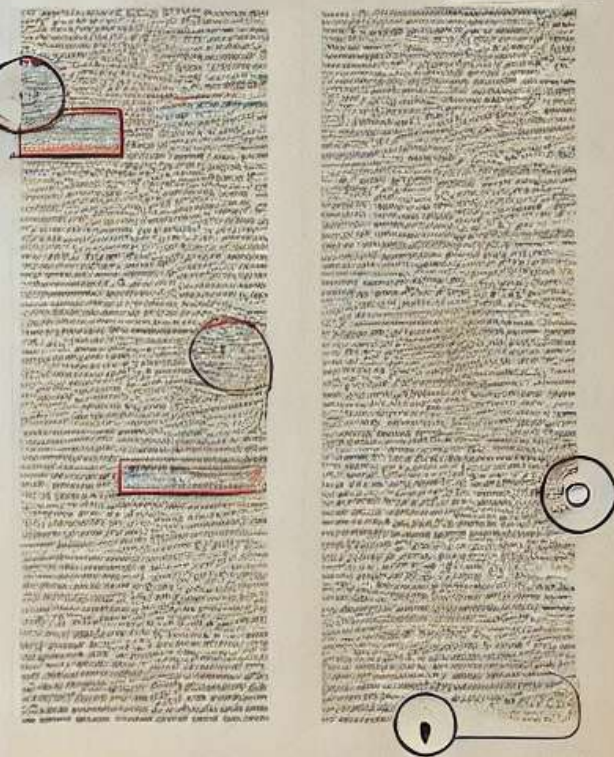
Key Advantages:

- Retains context from sources
- Reduces hallucinations
- References specific paragraphs
- Comparisons, summaries... even podcasts!

Use Cases:

- Technical paper analysis
- Literature reviews
- Class preparation
- Student guidance

## RESEARCH PAPER ANALYSIS



# Implementation

## Basic Workflow:

1. Upload documents (or links to YouTube videos!)
2. Automatic initial analysis
3. Ask specific questions
4. Extract insights

## Best Practices:

- Use structured questions
- Verify sources
- Balance exploration and deep dives
- Document findings

## Practical Examples:

- Extracting methodologies
- Comparing results
- Identifying gaps
- Summarizing for students

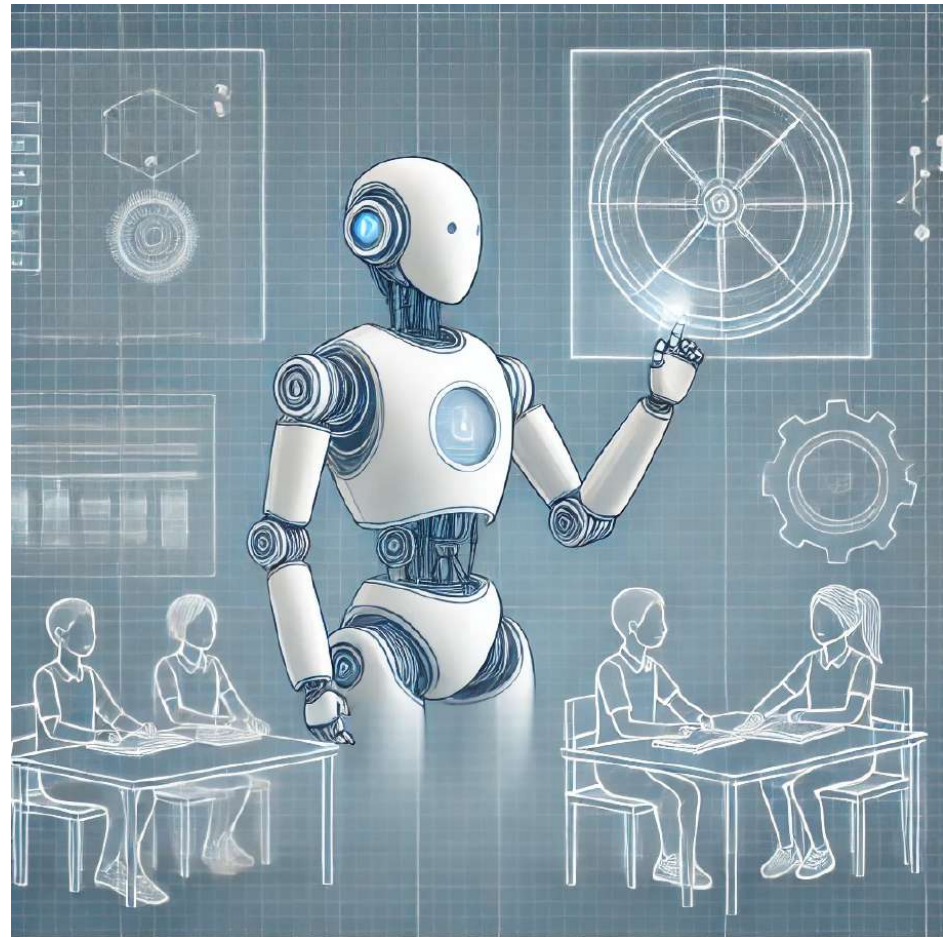
**Prompt Ejemplo:** Compare and contrast these papers' approaches and findings. Identify patterns, contradictions, and gaps that could inform future research.



# Practical exercise: Learning a new topic

## Steps to Get Started:

1. Open an account (<https://notebooklm.google.com/>)
2. Create a Notebook
3. Upload sources (e.g., four papers on a topic like quantum computing)
4. Experiment with different features



Notebook: <https://notebooklm.google.com/notebook/c973a477-6cdb-488e-a423-0dab55fe0386>  
Podcast: <https://notebooklm.google.com/notebook/c973a477-6cdb-488e-a423-0dab55fe0386/audio>



The background is a complex collage of scientific and academic illustrations. It features various charts and graphs, including a line graph, a bar chart, and a pie chart. There are also molecular structures, a DNA helix, and a glowing book in the foreground. The overall theme is research and knowledge.

# Perplexity: state of the art workflow

## Why Perplexity?

- AI-enhanced academic search
- Real-time, verifiable references
- Integration with bibliographic databases
- Synthesis from multiple sources

## Differences from Traditional Search:

- Synthesized answers
- Real-time references
- Guided exploration
- Automatic cross-verification

## Pro Features:

- Advanced models (GPT-4, Claude)
- Deep academic search
- Collections and organization tools
- Structured export options



# AI ASSISTANT Implementation

## Research Workflow:

1. Define the search.
2. Explore sources.
3. Synthesize information.
4. Organize results.

## Effective Strategies:

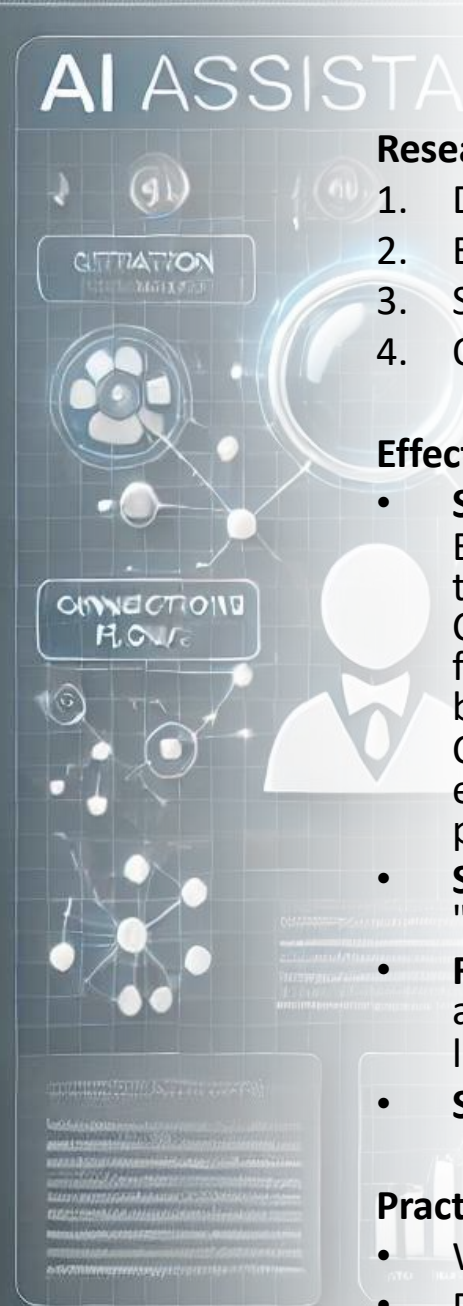
### Structured Searches:

Example: "Generate a short state-of-the-art review in Model Predictive Control, citing references in APA format. Use references published before 2018 from reputable journals. Cite at least 10 references exhaustively and academically in paragraph format."

- **Source Validation:** Use tools like "View Sources" to verify authenticity.
- **Reference Management:** Download and refine references, e.g., using tools like Gemini 1.5 Pro.
- **Select Academic Focus**

## Practical Applications:

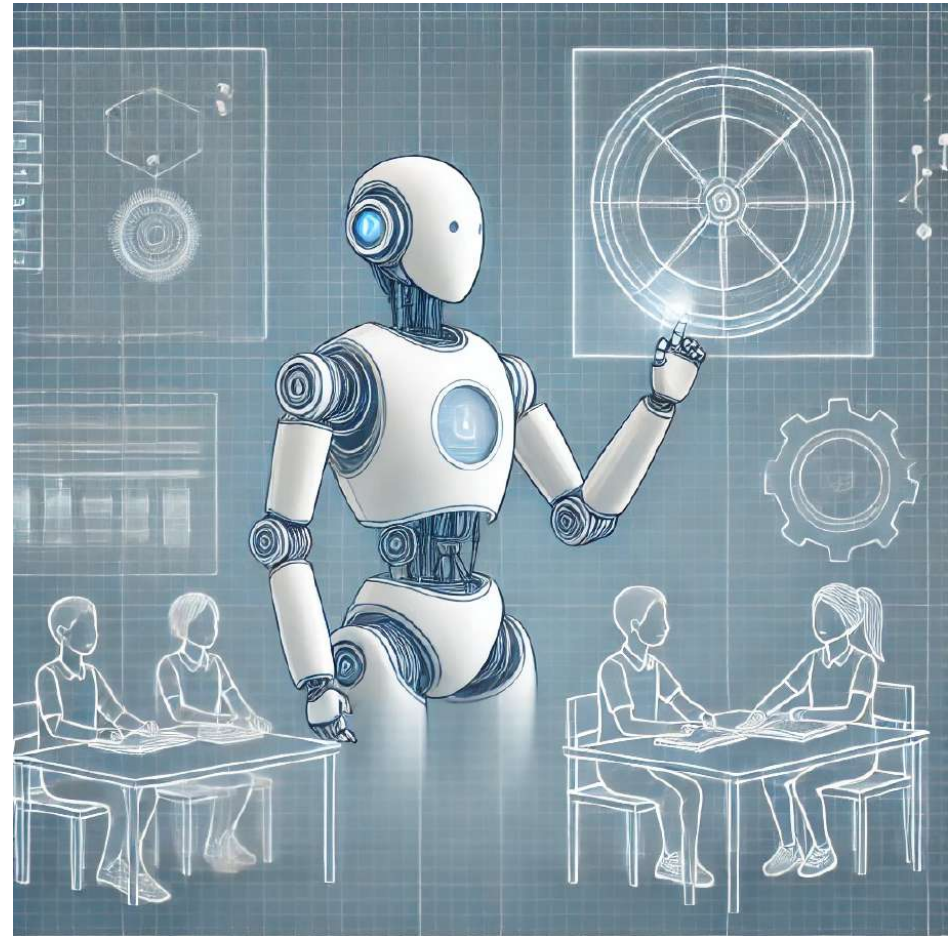
- Writing state-of-the-art reviews.
- Preparing class materials.
- Updating course content.
- Conducting guided research.





# Practical exercise: short state of the art

Adapted to your field



1. Generate a short state of the art in Model Predictive Control providing the references used at the end in APA format. Use references before 2018, that are from good journals. Cite the references in your state of the art. Give me everything in paragraph format. Use at least 10 references. Be exhaustive and academic.
2. Enlarge your review with more articles on practical applications, providing the references used at the end in APA format ... [same paragraph]
3. Can you give me the latex code of your response? <https://www.perplexity.ai/page/generate-a-short-state-of-the-H1Z2tb7jTLafbPajzvhxvg>

# Web Design

## Use Cases:

- Research project website
- Personal academic webpage/CV
- Conference or workshop website
- Research group portal
- Course or subject landing page
- Website for publishing results





# Implementation

## The Power of Prompts

### PREPARATION:

- Gather your actual content.
- Organize main sections.
- Clearly define requirements.

### CONTEXT AND ROLE:

"You are an expert web developer specializing in creating modern, responsive, and accessible websites using HTML and Tailwind CSS. I will provide real content for an [academic/research/personal] website. I need you to:

1. Write clean, well-structured code.
2. Use only Tailwind CSS core classes (no arbitrary values).
3. Ensure responsive design.
4. Follow accessibility principles.
5. Only use the content I provide."

### BASE GENERATION:

#### Prompt:

"Create a web structure for a research project on [TOPIC] with these sections: [YOUR SECTIONS]"

### ADDING CONTENT:

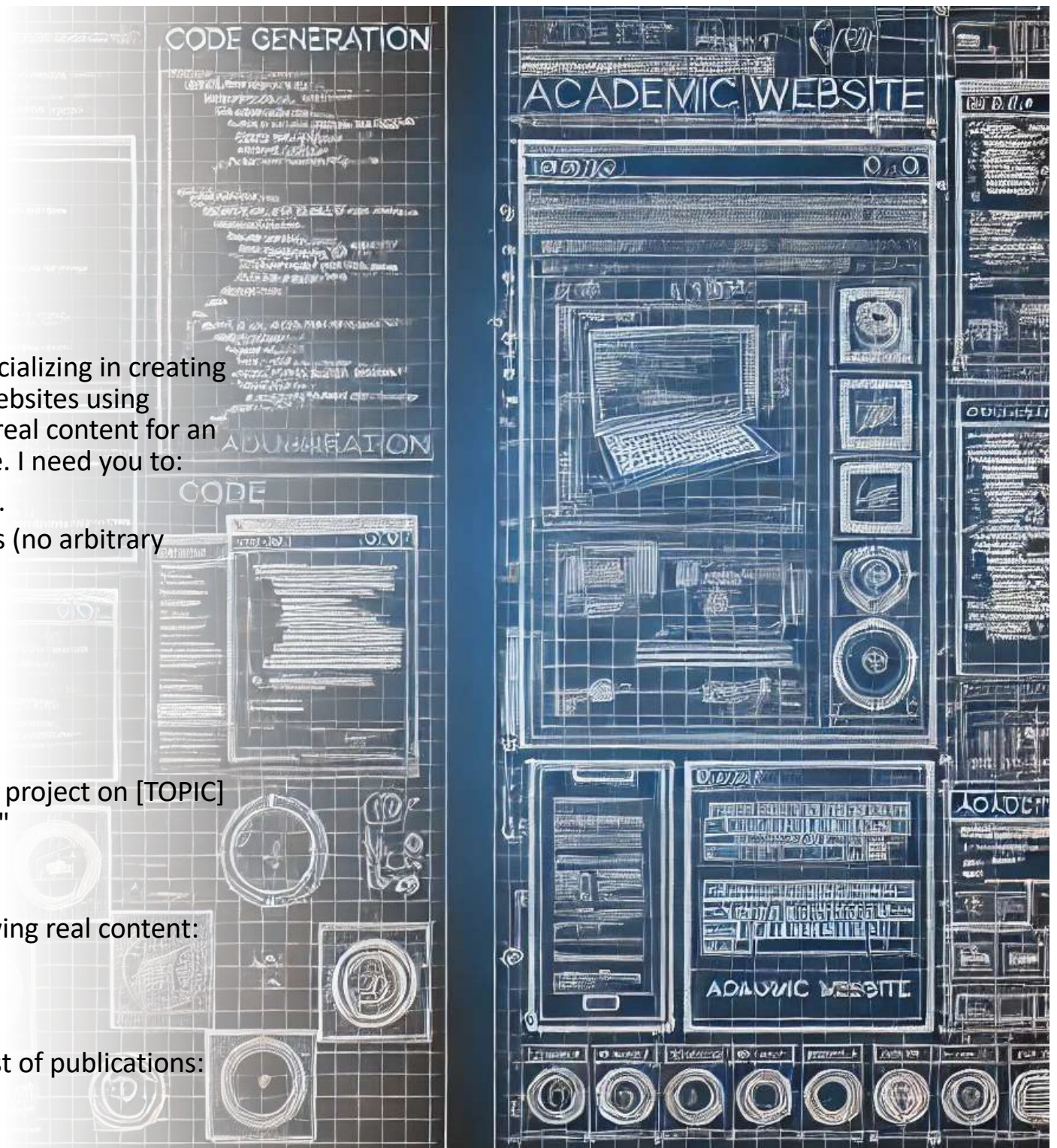
#### Prompt:

"Modify this HTML to insert the following real content: [PASTE YOUR CONTENT]"

### SPECIFIC COMPONENTS:

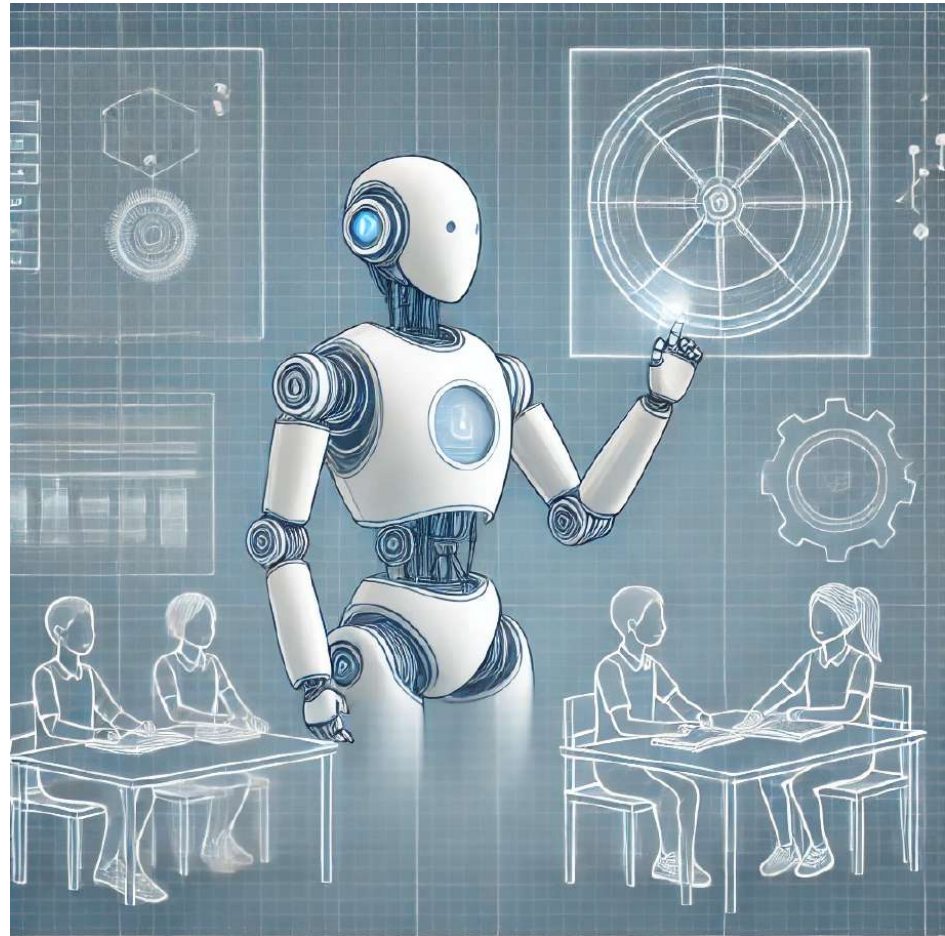
#### Prompt:

"Create a component to display this list of publications: [YOUR PUBLICATIONS]"





# Exercise: page for a research project



## Practicing Prompts

**Content:**Provide a project proposal.

**Ask for Additional Information:** Prompt the AI to identify missing details and supply the requested information.

**Use Placeholders for Images:** Include placeholders where visuals will be added later.

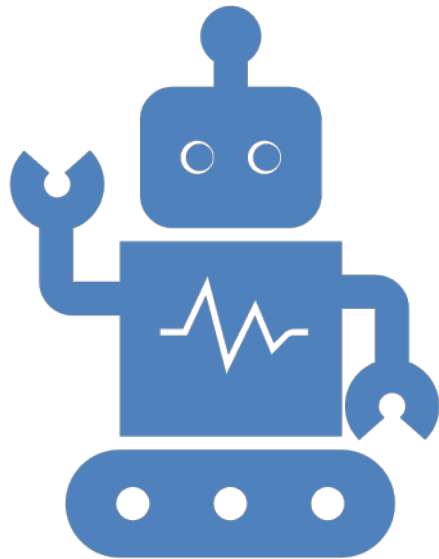
**Add a List of Papers**

**Iterate if Necessary**

<https://aero.us.es/rvazquez/TallerIA/project-website-1.html>

<https://aero.us.es/rvazquez/TallerIA/project-website-2.html>

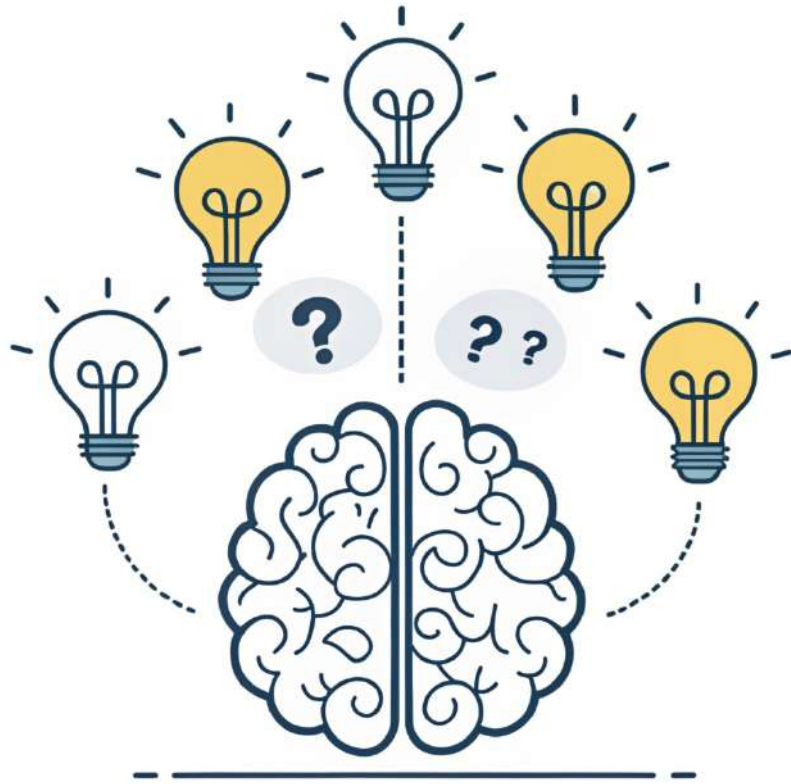
<https://aero.us.es/rvazquez/TallerIA/project-website-3.html>



# Outline

1. A personal journey into AI
2. How LLMs work
3. Basic Prompt Engineering
4. Updated LLM Ecosystem (Dec 2024)
5. Limitations of LLMs
6. Advanced Prompt Engineering
7. Generating documents
8. Generating visual presentations
9. Generating code
10. Other use cases and tools
- 11. Closing and discussion**



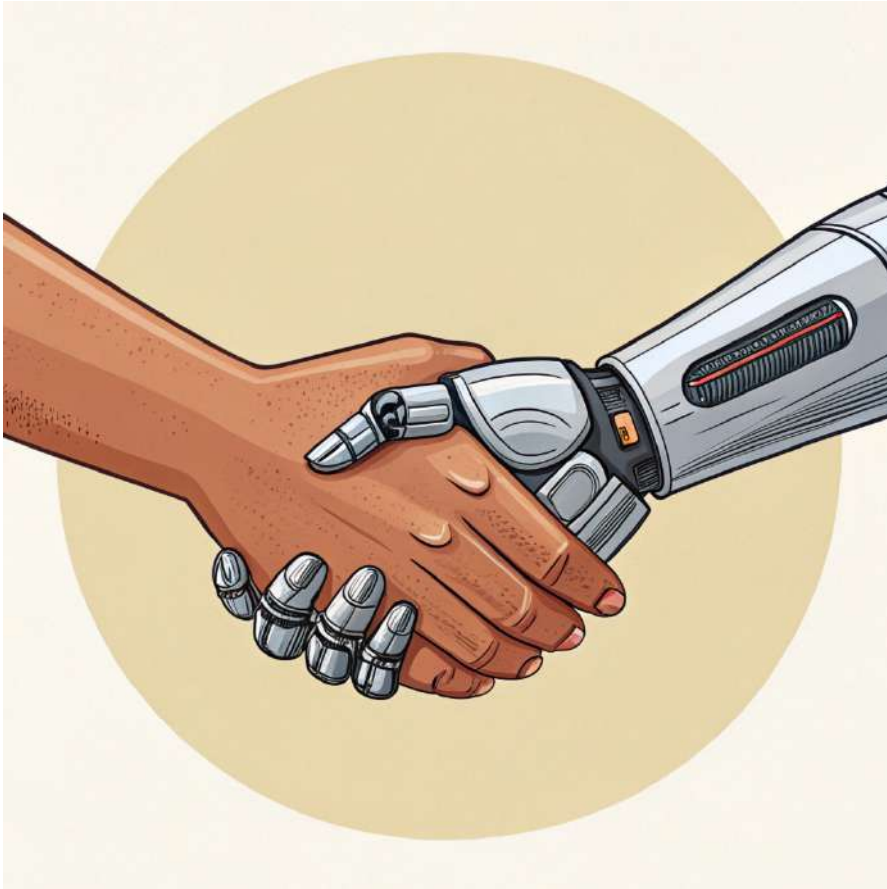


# Conclusions

## *The Art of Asking Questions*

**Talking with AIs is the new soft skill of the 21st century. It's not enough to know them; you must develop the art of using them.**

- Don't teach how to use AI; teach how to think with AI.
- The best question is worth more than a thousand answers.
- Take advantage of AI: for teaching + research + augmented management.



## Conclusions

### *The Speed of Change*

**Technology evolves at an exponential pace, but the competitive advantage lies in combining the power of machines with human wisdom.**

- Your teaching experience is your greatest asset: AI doesn't know your students.
- Don't compete with AI; integrate its power into your methodology.
- Build bridges between academic tradition and technological innovation.



# Conclusions

## *The Future is Now*

**Artificial Intelligence will not replace us; we will be replaced by those who know how to use it better than we do.**

- Be a pioneer in all three areas: research better, teach differently, and manage more efficiently.
- Build an ecosystem where AI enhances every aspect of your academic work.



# Conclusions

## *Call to Action*

**Lead by example: academia is the perfect ground for fostering responsible innovation.**



# A final survey







# Let's surf together the AI tsunami!



[rvazquez1@us.es](mailto:rvazquez1@us.es)



Créditos: Rafael Vazquez y Claude 3.5 Sonnet New, ChatGPT 4o, ChatGPT o1, Recraft.ai  
Gemini 1.5 Pro 002, DALL-E 3, ideogram.ai, HeyGen.com, Udio.com, Elevenlabs.com

