Information Technology Institute

Generative AI tools And prompt engineering

Hany Saad
hghiett@iti.gov.eg

Course Outline

Course Title: Intro to Generative AI and prompt engineering

Tracks: All tracks (Including Development tracks)

Duration: 6 hours (Lecture + workshop)

Evaluation:

- Attendance + Report / presentation.

Prerequisites:

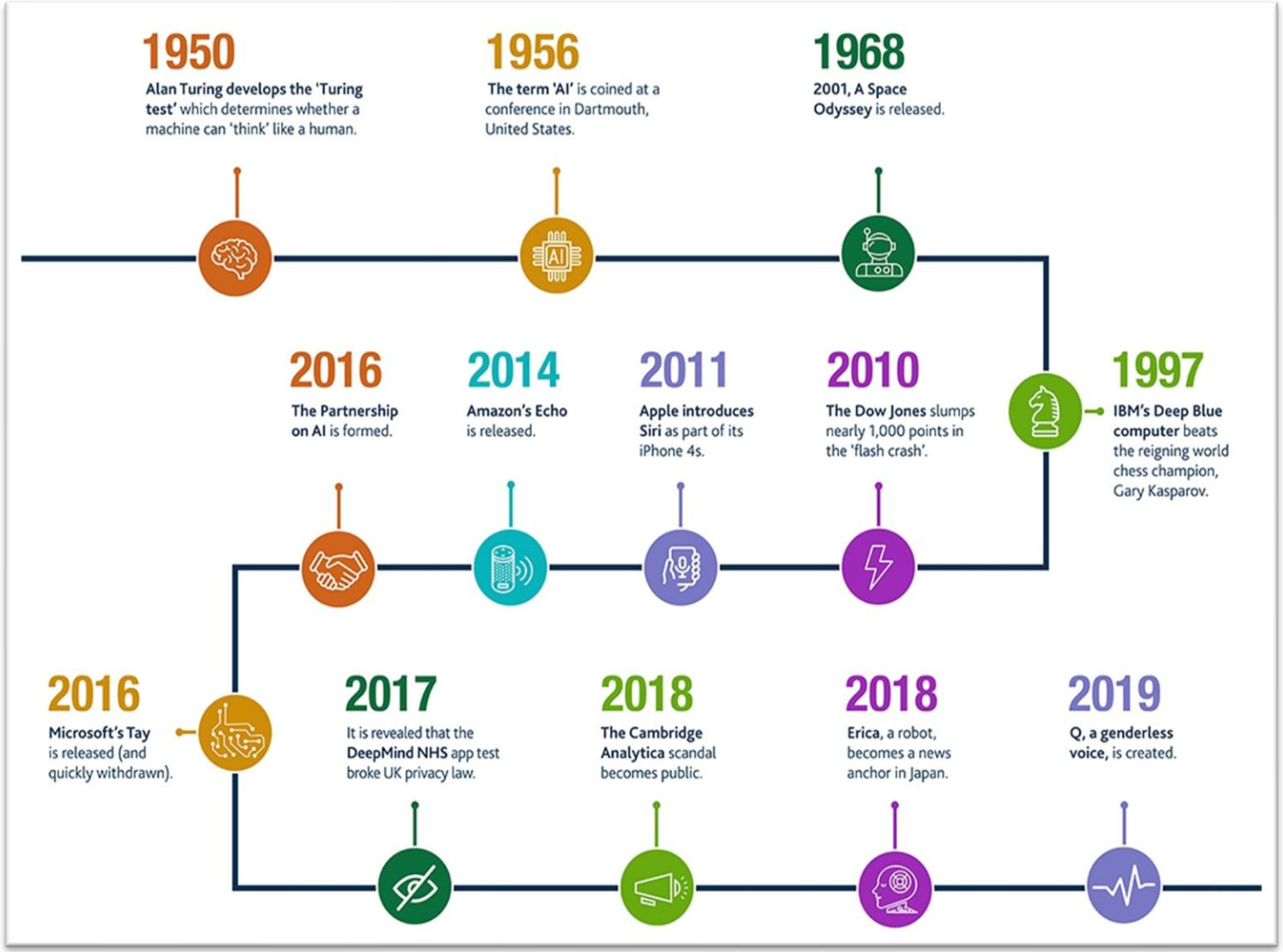
- No prerequisites, preferred to be in the last month.

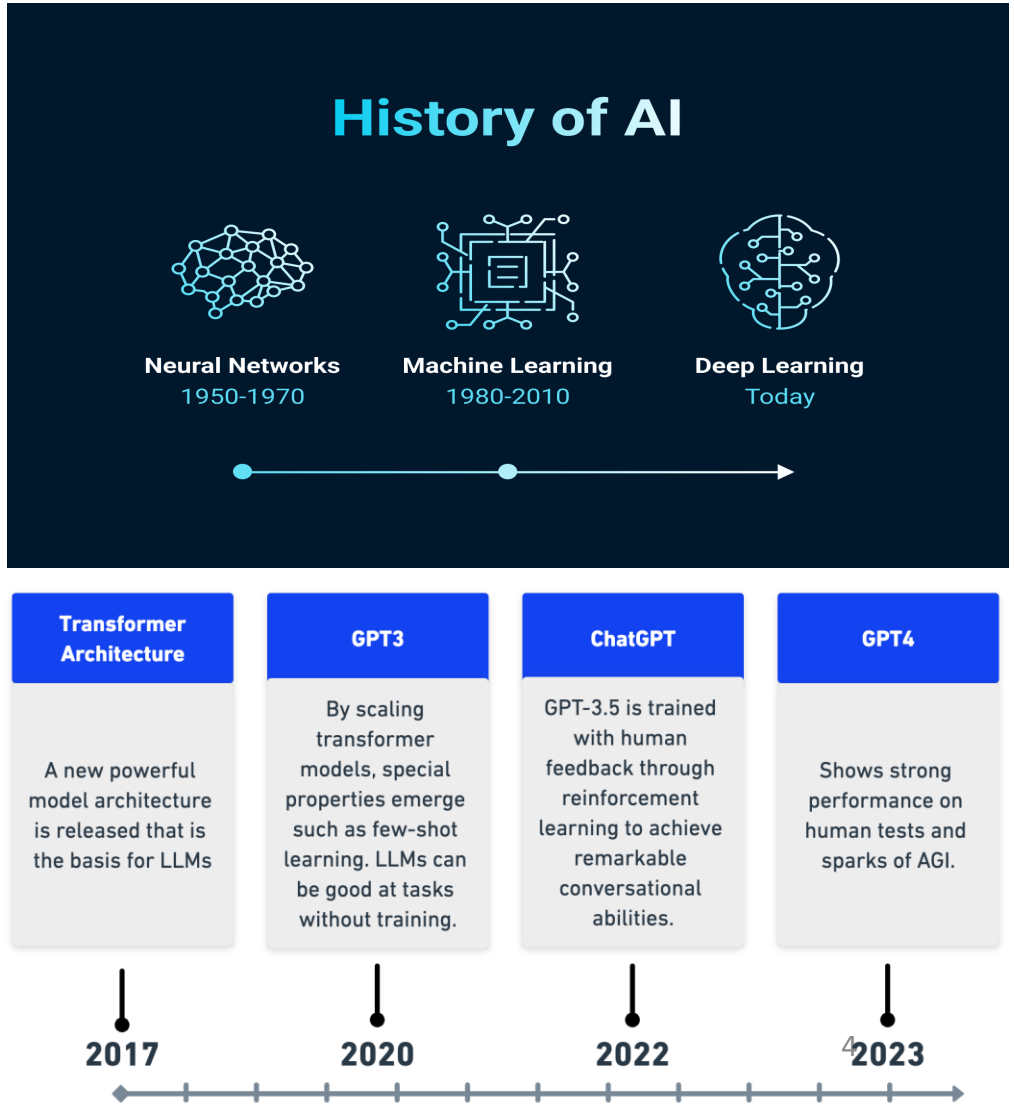
Outlines:

- Intro to Gen AI: evolution of AI to Generative AI.
- Gen AI benefits and practical use cases
- Key Gen AI terminologies and merging technologies: LLM, Prompts & completions, Tokens, Models, Multi-modal models, RAG, Fine-tuning, Reasoning models, AI agents.
- Gen AI tools models, tools (ChatGPT, Gemini, Claude, MS Copilot, DeepSeek, Qwen, Grok…) and use cases in different fields.
- Introducing prompt engineering
- Use cases for Gen AI applications and uses (to be tailored for each track)
- AI tools limitations, security and data privacy considerations

Why do we need to know about Gen AI?

History of AI



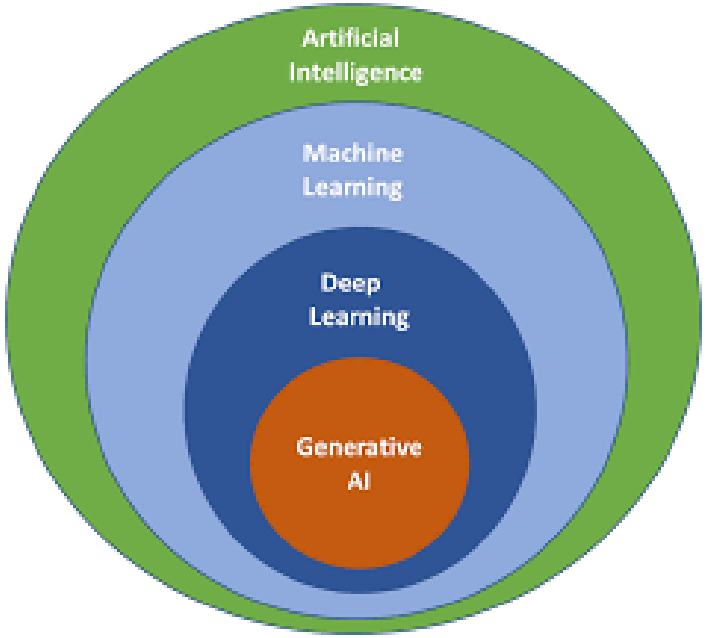


What's Generative AI?

Generative AI: is a type / subset of artificial intelligence system that can produce various types of content, including text, imagery, audio and synthetic data.
Generative AI models learn the patterns and structure of their input training data and then generate new data that has similar characteristics.

Generative AI Tools

- **Text**
  - text generation
- **Visual**
  - image generation
  - video generation
  - design generation
- **Audio**
  - voice generation
  - music generation
- **Code**
  - code generation

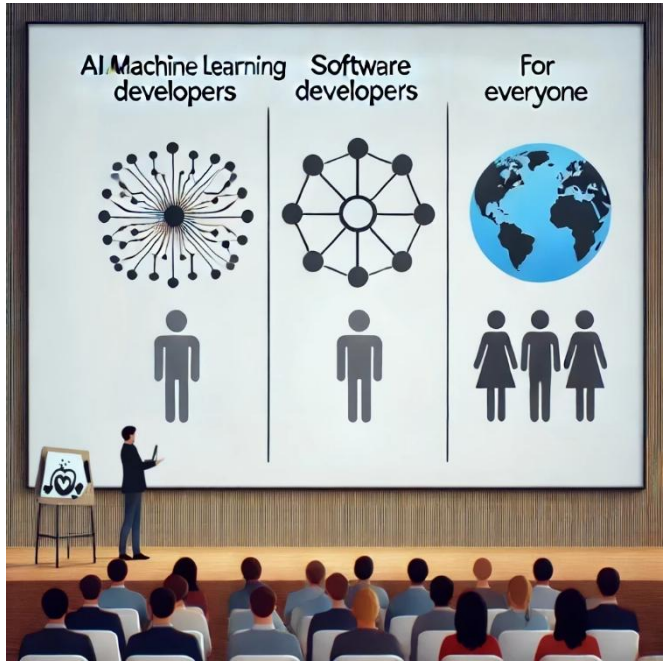


Who should learn about Gen AI?

- **AI / Machine Learning Developers**:
  - Design and refine Generative AI models to create more intelligent and efficient AI-driven solutions.

- **Software Developers**:
  - Utilize AI tools to enhance productivity in development tasks.
  - Leverage Generative AI APIs to build AI-powered applications

- **Everyone**:
  - In today's AI-centric world, it's crucial for individuals across all professions to grasp the basics of Generative AI and incorporate AI tools into their workflows to boost productivity.

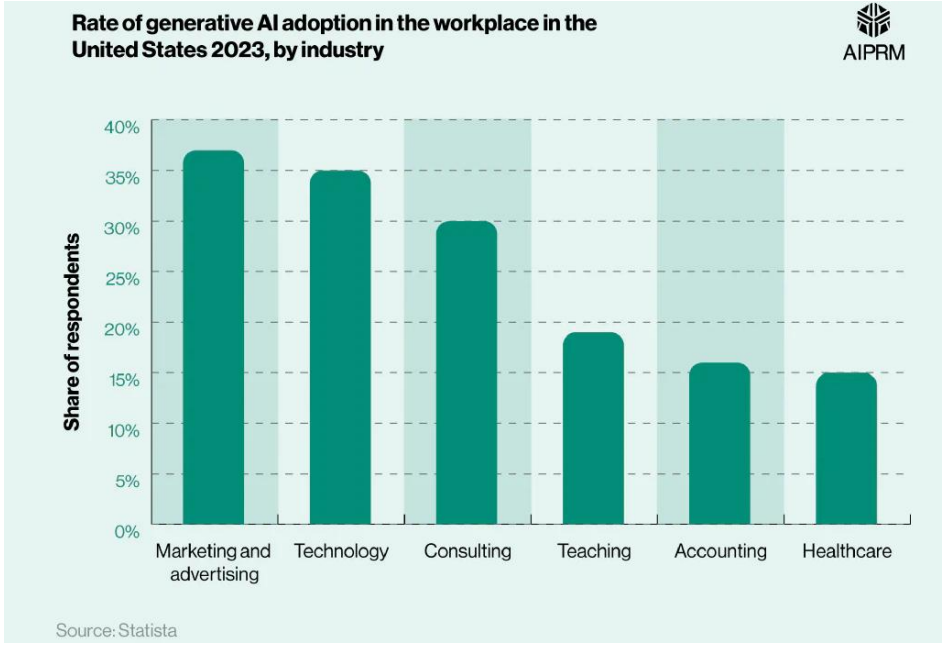


Will AI Replace Jobs / employees?

NEWS

As many as 41% of employers plan to use AI to replace roles—but it's not a 'jobs apocalypse,' experts say

By Ryan Johnston, CNBC • Published February 26, 2025 •



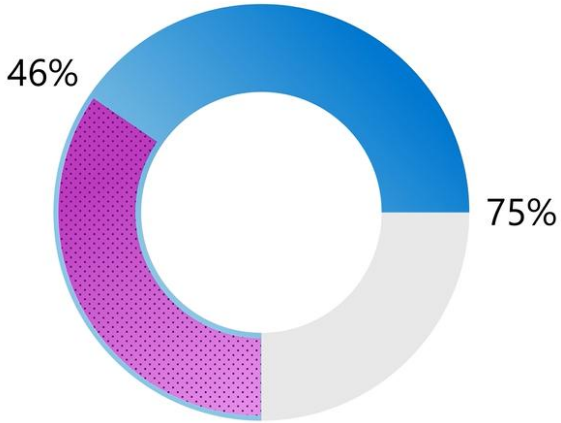
Gen-AI Tools in Work Statistics

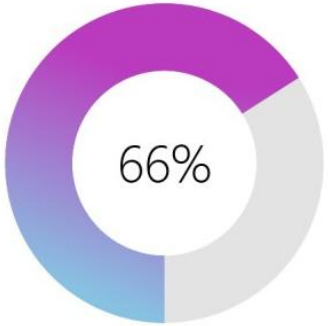
Three Out of Four People Use AI at Work

Usage nearly doubled in the last six months.

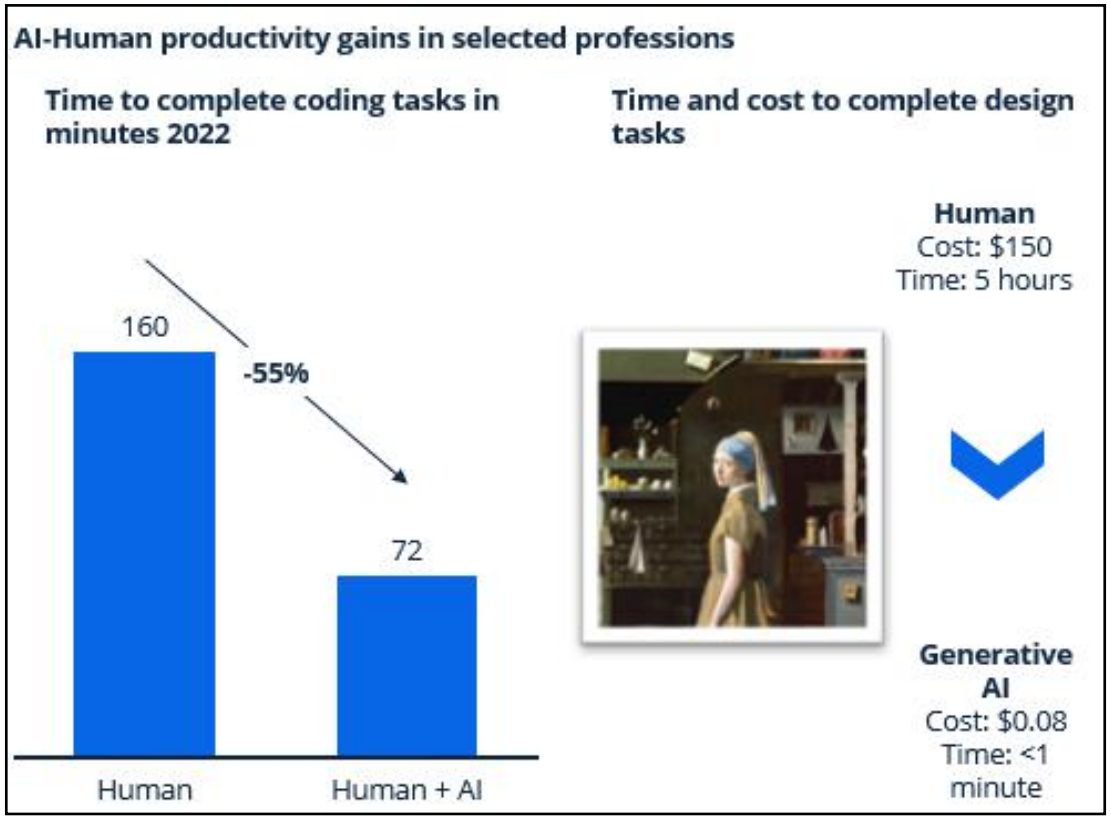
75% of people are already using AI at work

46% of them started using it less than 6 months ago





Leaders would not hire someone without AI skills

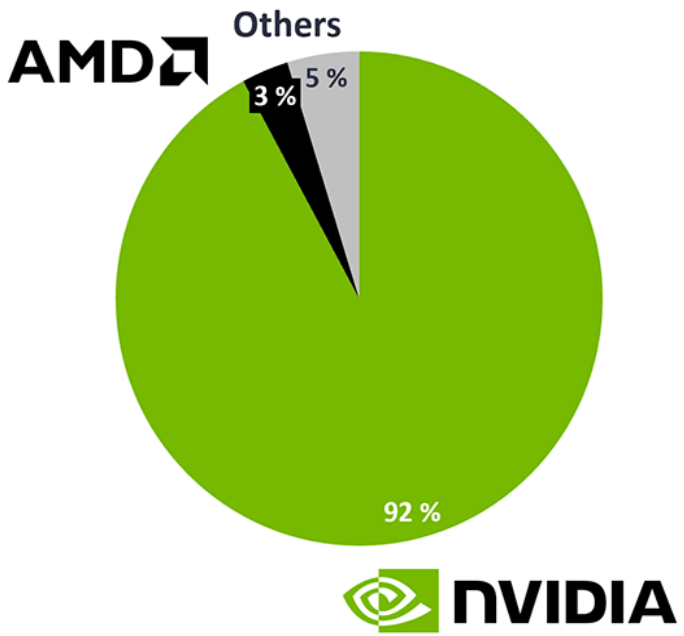


2024 Work Trend Index Annual Report from Microsoft and LinkedIn: AI In Work

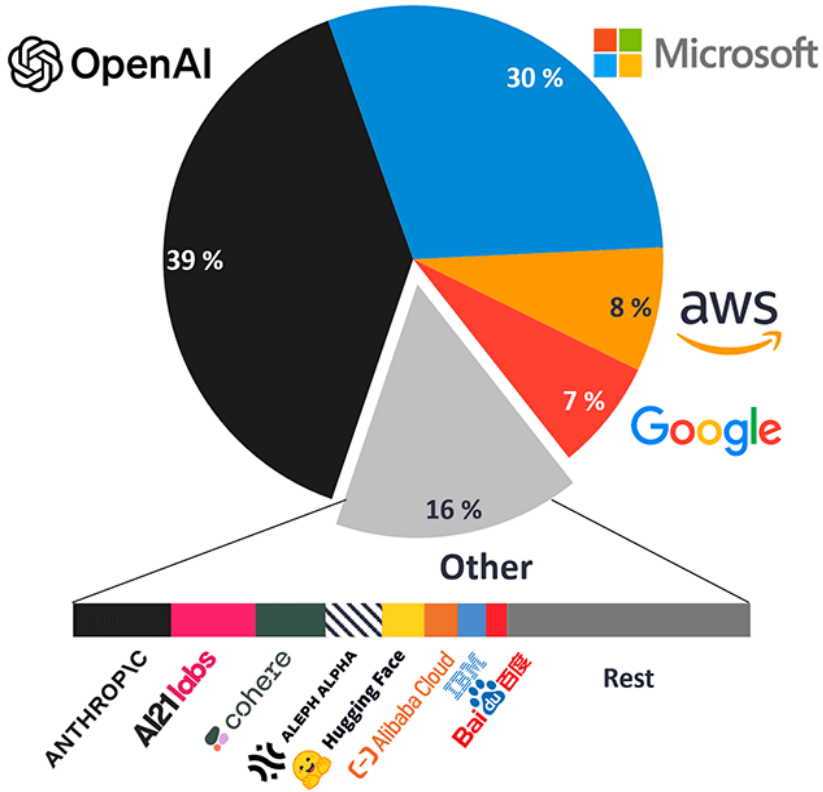
Nielsen Norman Group: AI Improves Employee Productivity by 66%

Generative AI: Market share of leading vendors 2023

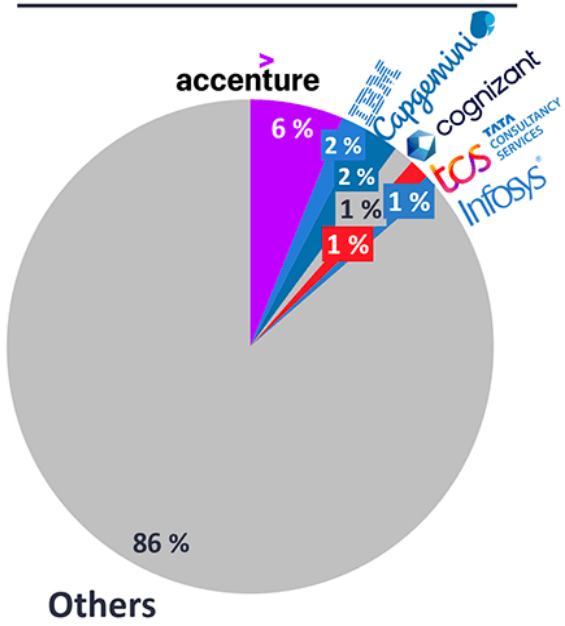
1 Data center GPUs



2 Models & platforms



3 Services



Note: Numbers are rounded and might not add up to 100%; Market share is based on 2023 market sizes (based on revenue).
Source: IoT Analytics Research 2023-Generative AI Market Report 2023-2030. We welcome republishing of images but ask for source citation with a link to the original post and company website.

Sample GenAI Freelancing jobs

- **Upwork**
  - AI Image Generation and Deepfake Model Creation
  - AI developer (rag + llm) + fullstack
  - Next.js Developer Needed for SaaS Application Integration with AI Features
  - Developers for 5 AI agents
  - AI Engineer for Medical RAG-based Chatbot Development
  - GenAI engineer to build GenAI-powered modules of an astrology app
  - CReate AI Agent using Open AI or Google AI API or CrewAI
  - Full-Stack Developer with AI Experience (Replit Preferred)
  - Developer Needed to Build AI Agents for Data Analysis and Digital Advertising
  - Web App Development with AI for Meta Ads Management
  - … and more
- **Khamsat and Mostaql**
  - Khamsat Gen AI Services
  - Mostaql Gen AI Projects

Gen-AI based startups and real products

- **Use cases for real products using Gen AI:**
  - Loop-x (AI Agents for different roles): https://loop-x.co
  - Gemelo ai (Twin AI creator): https://gemelo.ai
  - Apriora AI (Recruitment and Interviews): https://www.apriora.ai
  - Velents AI (Recruitment and Interviews): : https://www.velents.com
  - Nancy AI (Recruitment and Interviews): : https://nancy-ai.com
  - Mariana AI (Medical): https://marianaai.com

How to use the AI tools responsibly?

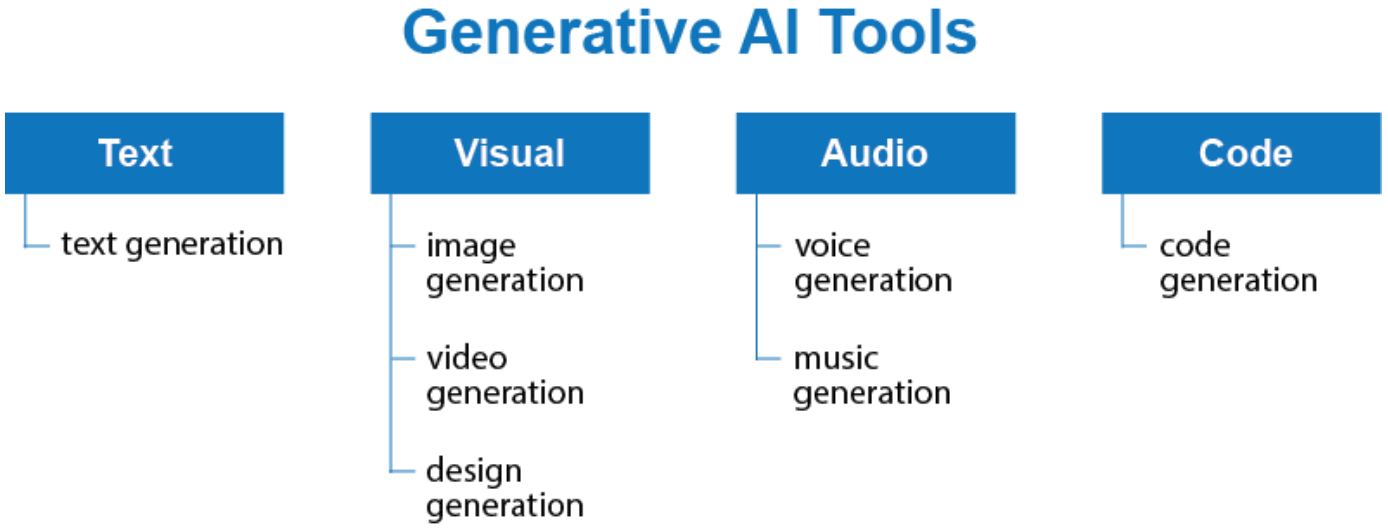
- **As a Student/Learner:**
  - ✅ **Use AI for:** Asking for explanations, self-testing, researching, and similar activities.
  - ❌ **Don't use AI for:** Assignments, projects, tasks, or topics you're still learning.
  - **Build your skills** to work effectively both with and without AI.

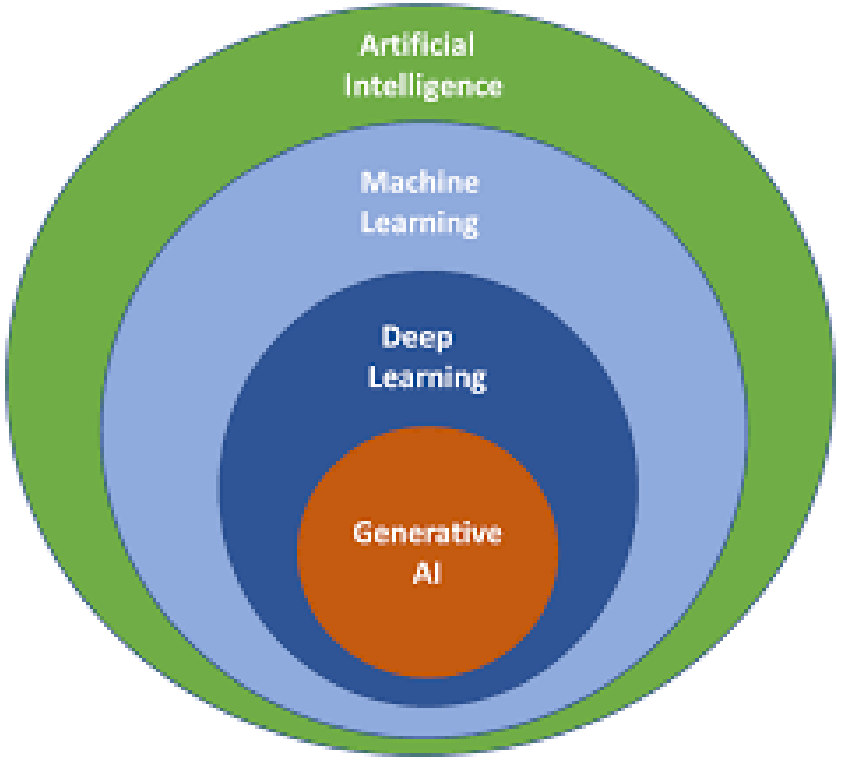
- **In a Job or Freelancing:**
  - **Check first** if AI tool usage is permitted and which tools are allowed.
  - **Use AI to boost productivity** and improve your work.
  - **Always review** AI-generated content carefully.
  - **Don't rely on AI** for tasks you have no prior knowledge of.

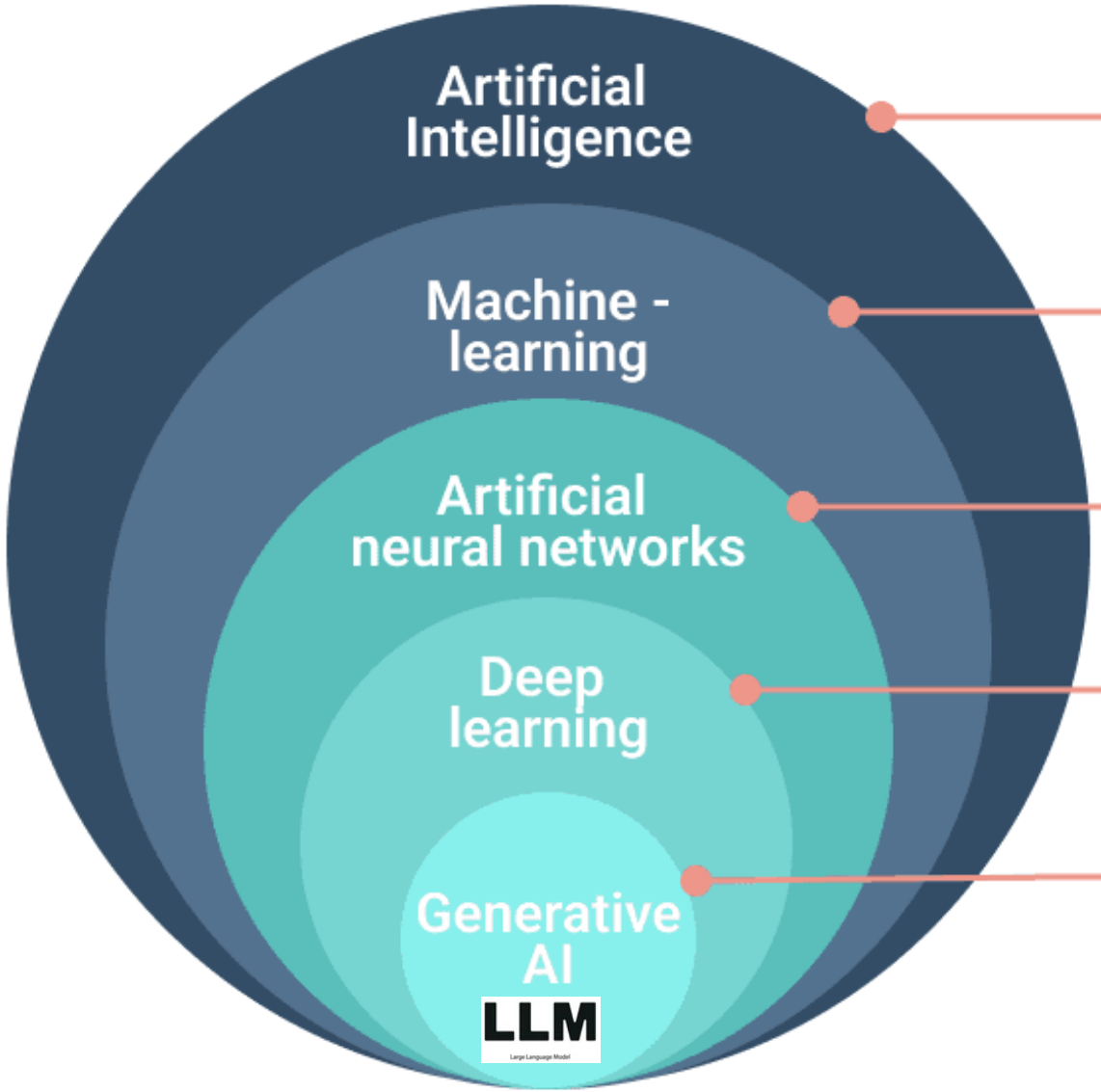
Introducing Generative AI

What's Generative AI?

Generative AI: is a type / subset of artificial intelligence system that can produce various types of content, including text, imagery, audio and synthetic data.
Generative AI models learn the patterns and structure of their input training data and then generate new data that has similar characteristics.







Artificial Intelligence (AI)

Development of machines that can perform tasks requiring human-like intelligence.

Machine-learning (ML)

Training of computers to learn and make predictions or decisions without being explicitly programmed.

Artificial neural networks (ANN)

Human brain-inspired machine-learning models.

Deep learning (DL)

Neural networks with multiple layers to extract complex patterns and make accurate predictions or decisions.

Generative AI (GenAI)

Creating models and systems that can generate new and original content.

Image 2. AI subsets

AI terms

- **A Large Language Model (LLM):** is an advanced AI system trained to understand and generate human-like text. It learns from massive amounts of text data and can answer questions, write content, and assist in conversations.
- **AI model:** is a system trained to recognize patterns, make predictions, and automate tasks based on data. It learns from examples and improves its performance over time..
- **Multimodal models:** can process a wide variety of inputs, including text, images, and audio, as prompts and convert those prompts into various outputs, not just the source type.
- **Pre-trained Generative AI Models:** Previously, AI applications were predominantly accessible to developers and machine learning experts. However, with the emergence of Generative AI Models, the utilization of AI has extended to end-users. Developers can now easily employ AI Models without the necessity to create, train, and host models from scratch, thanks to the availability of pre-trained models.
- **Reasoning Model** is a type of AI system designed to simulate human logical thinking. It processes information through a set of rules or logic to draw conclusions, solve problems, or make decisions.
- **Deep Thinking** is the ability to analyze complex problems, explore multiple perspectives, and make well-reasoned decisions. It involves **critical analysis, logical reasoning, and reflection** rather than relying on surface-level understanding.
- **Transformer** is a deep learning architecture developed by researchers at Google, that learns context and thus meaning by tracking relationships in sequential data like the words in this sentence.

AI terms (Cont.)

- **Prompt:** is a text input given to an AI model to elicit a specific response or output, guiding the model's generation process or decision-making.

- **Completion:** is the output or response generated by an AI model based on a given prompt, representing the model's attempt to complete, answer, or extend the input it received.

- **Tokens:** For text tokens Azure OpenAI processes text by breaking it down into tokens. For example: 1,000 tokens is about 750 words.

- **Prompt engineering** is the process of structuring an instruction that can be interpreted and understood by a generative AI model. A prompt is natural language text describing the task that an AI should perform.

AI terms (Cont.)

- **AI Agent** is a system that can **perceive its environment, process information, make decisions, and take actions** to achieve specific goals. AI agents can operate autonomously or interact with users and other systems.

- **Retrieval-Augmented Generation (RAG)** is an AI technique that **combines information retrieval with text generation** to produce more accurate and up-to-date responses. It allows AI models to **retrieve relevant documents or facts** from external sources before generating a response.

LLM evolutionary tree

Source: https://github.com/Mooler0410/LLMsPracticalGuide

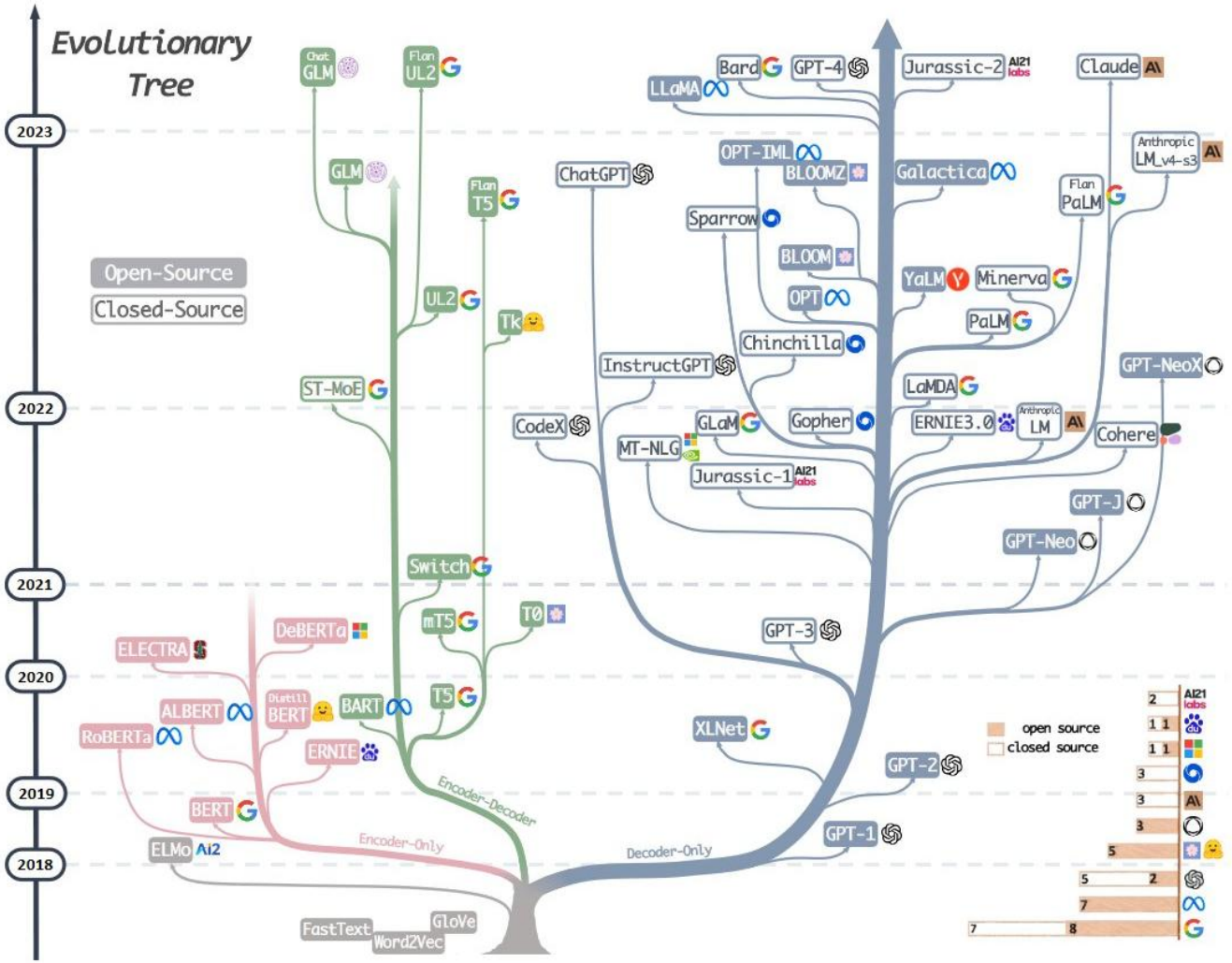
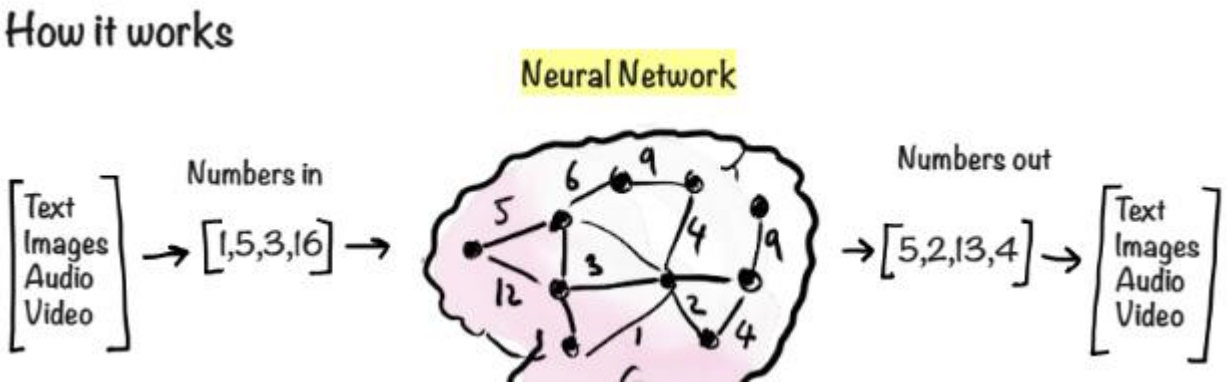
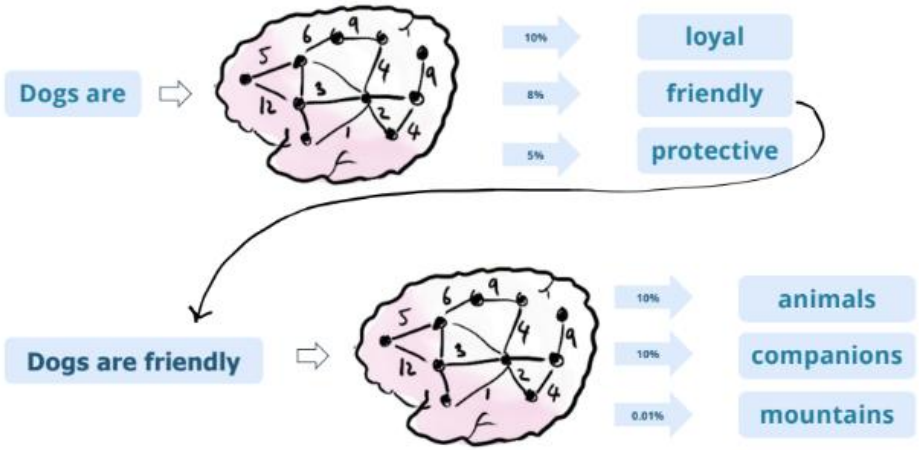


Fig. 1. The evolutionary tree of modern LLMs traces the development of language models in recent years and highlights some of the most well-known models. Models on the same branch have closer relationships. Transformer-based models are shown in non-grey colors: decoder-only models in the blue branch, encoder-only models in the pink branch, and encoder-decoder models in the green branch. The vertical position of the models on the timeline represents their release dates. Open-source models are represented by solid squares, while closed-source models are represented by hollow ones. The stacked bar plot in the bottom right corner shows the number of models from various companies and institutions.

How it works?





Dogs are friendly

Dogs are friendly animals

Dogs are friendly animals who

Dogs are friendly animals who love

Dogs are friendly animals who love to

Dogs are friendly animals who love to wag

Dogs are friendly animals who love to wag their

Dogs are friendly animals who love to wag their tails

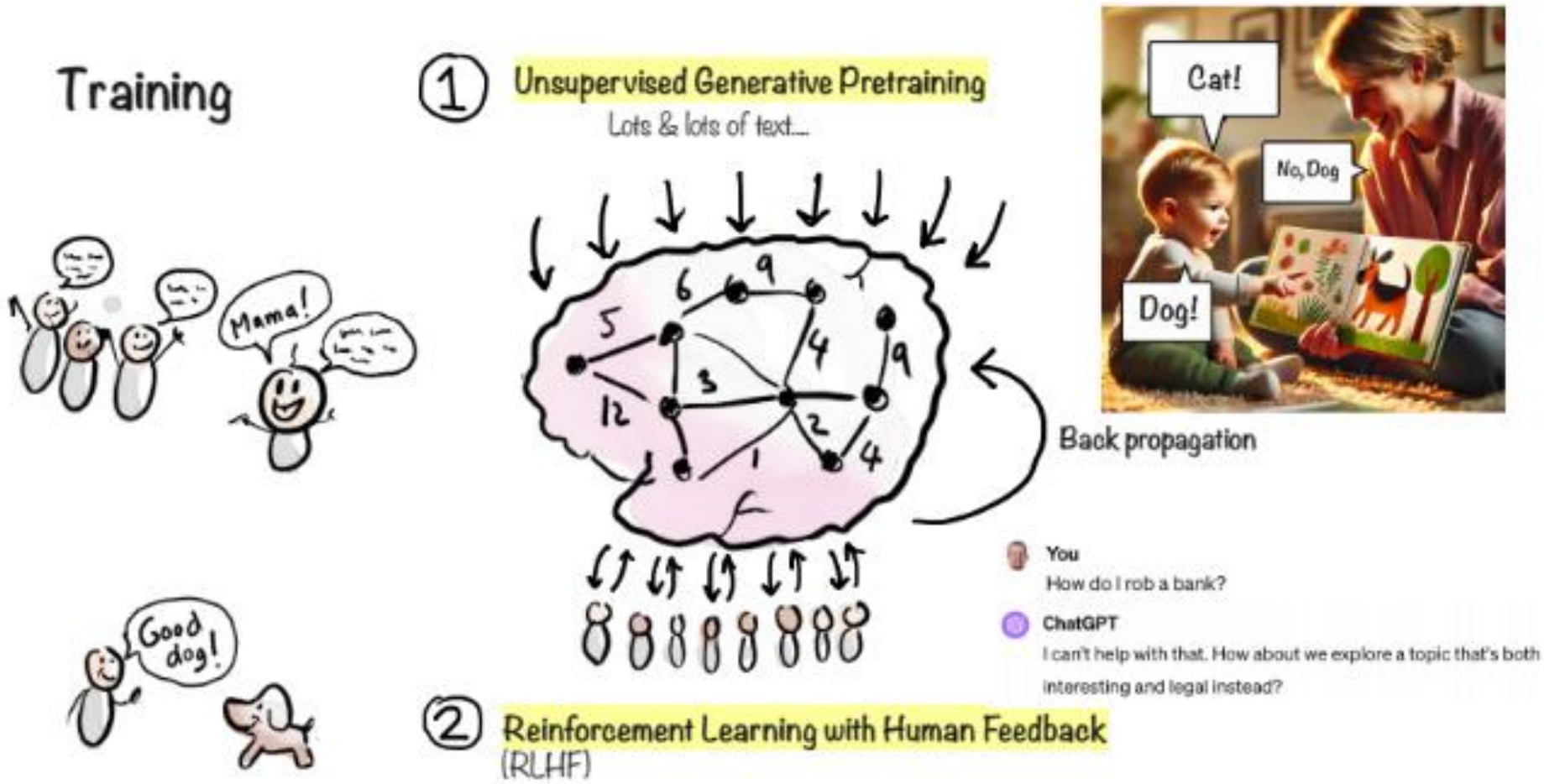
Dogs are friendly animals who love to wag their tails and

Dogs are friendly animals who love to wag their tails and make

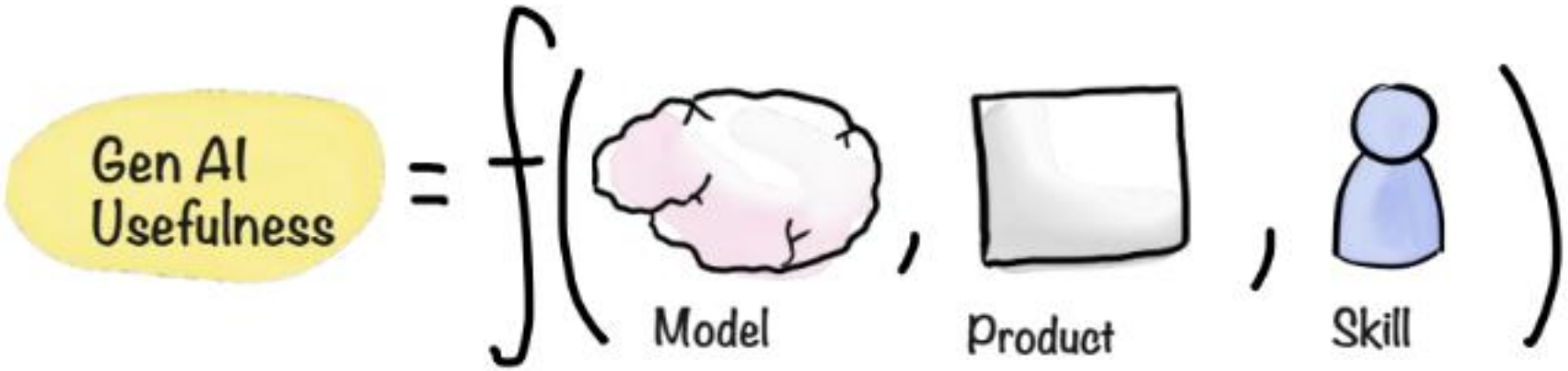
Dogs are friendly animals who love to wag their tails and make new

Dogs are friendly animals who love to wag their tails and make new friends.

How it works? (Cont.)



Gen AI Usefulness



Seem really good, but haven't tried so much myself:
- gemini pro
- grok
- deepseek

Mind blowing:
- o1 pro, o3 (advanced research)

Awesome:
- claude-3.5-sonnet

Pretty good:
- gpt-4o

Good:
- ChatGPT
- Claude
- Cursor (coding)
- Perplexity (research)
- Github Copilot (coding)

Bad:
- MS Copilot

- Use cases
- Prompt Engineering
- How to use the products
- Weaknesses / limitations
- Which models are good for what

Henrik Kniberg Ymnig.ai

Models and Tools

OpenAI Models

Models - OpenAI API

Models overview

The OpenAI API is powered by a diverse set of models with different capabilities and price points. You can also make customizations to our models for your specific use case with fine-tuning.

MODEL	DESCRIPTION
GPT-4o	Our high-intelligence flagship model for complex, multi-step tasks
GPT-4o mini	Our affordable and intelligent small model for fast, lightweight tasks
o1-preview and o1-mini	Language models trained with reinforcement learning to perform complex reasoning.
GPT-4 Turbo and GPT-4	The previous set of high-intelligence models
GPT-3.5 Turbo	A fast, inexpensive model for simple tasks
DALL·E	A model that can generate and edit images given a natural language prompt
TTS	A set of models that can convert text into natural sounding spoken audio
Whisper	A model that can convert audio into text
Embeddings	A set of models that can convert text into a numerical form
Moderation	A fine-tuned model that can detect whether text may be sensitive or unsafe

Sora: Text-to-video model

- **Sora:** OpenAI's text-to-video model, it is an AI model that can create realistic and imaginative scenes from text instructions.

GPT-o1: OpenAI's New Reasoning Model

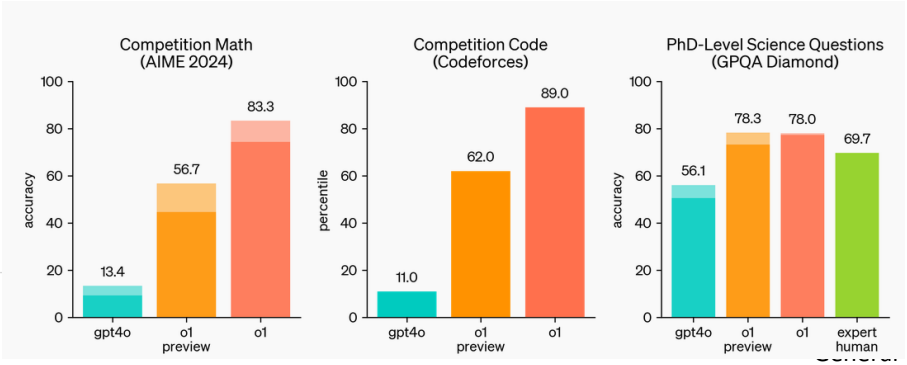
What is GPT-o1?

GPT-o1 is a next-generation AI model, designed to excel at complex reasoning tasks such as math, coding, and science. Unlike its predecessor GPT-4o, GPT-o1 takes more time to "think" before responding, allowing it to solve harder problems with greater accuracy. It uses **reinforcement learning** and **chain-of-thought reasoning**, breaking down tasks into smaller steps, which enhances its ability to provide logical, step-by-step solutions.

OpenAI o1 ranks in the **89th percentile** on competitive programming questions (Codeforces), places among the top 500 students in the US in a qualifier for the USA Math Olympiad (AIME) and **exceeds human PhD-level** accuracy on a benchmark of physics, biology, and chemistry problems (GPQA).

Chain of Thought

Similar to how a human may think for a long time before responding to a difficult question, o1 uses a chain of thought when attempting to solve a problem. Through reinforcement learning, o1 learns to hone its chain of thought and refine the strategies it uses. It learns to recognize and correct its mistakes. It learns to break down tricky steps into simpler ones. It learns to try a different approach when the current one isn't working. This process dramatically improves the model's ability to reason.



Top AI models and Chat-based AI tools

Gemini

ChatGPT

Claude

Meta AI

Grok

deepseek

Poe

perplexity

Qwen

Microsoft Copilot

AI tools Capabilities

AI tools capabilities and use cases:

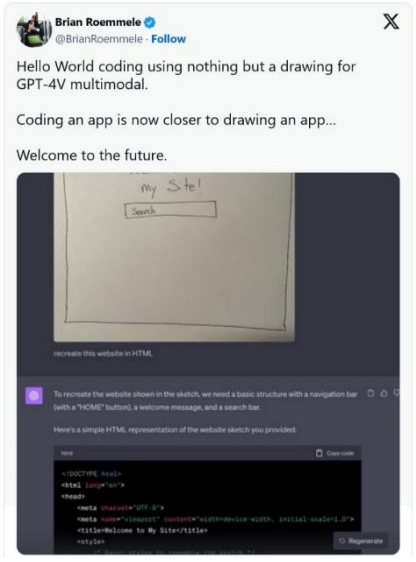
- **Text generation:** can generate text based on a given prompt, including full sentences, paragraphs, and even articles.
- **Conversational AI:** can be used to build chatbots and conversational interfaces, allowing it to handle various types of user inputs, such as natural language questions and commands.
- **Sentiment Analysis:** can determine the sentiment of a given text, whether it is positive, negative, or neutral.
- **Translation:** can translate text from one language to another, supporting multiple languages.
- **Question Answering:** can provide answers to questions asked in natural language.
- **Content creation:** can be used to generate content, such as articles, stories, and poems.
- **Text Classification:** can classify text into different categories, such as topics, genres, and sentiments.
- **Code Generation:** can generate code snippets based on given natural language descriptions.
- **And more…**

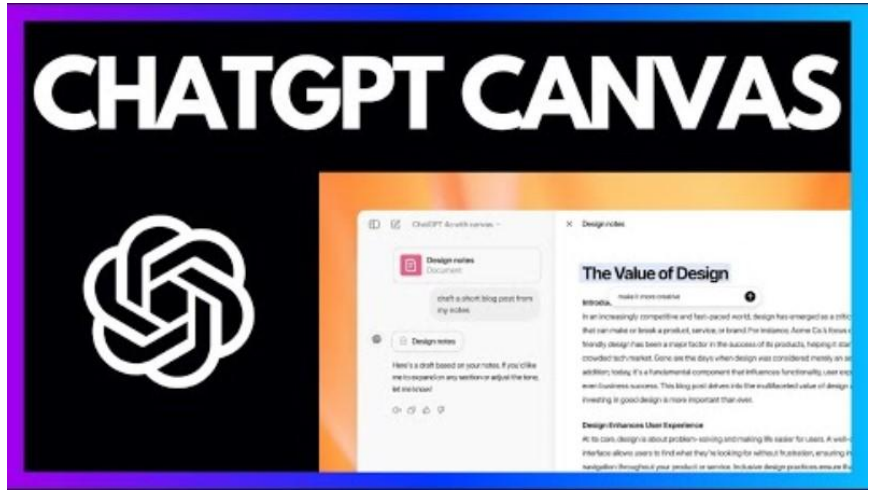
ChatGPT features and use cases

Exploring ChatGPT - as one of the most known tools - features:

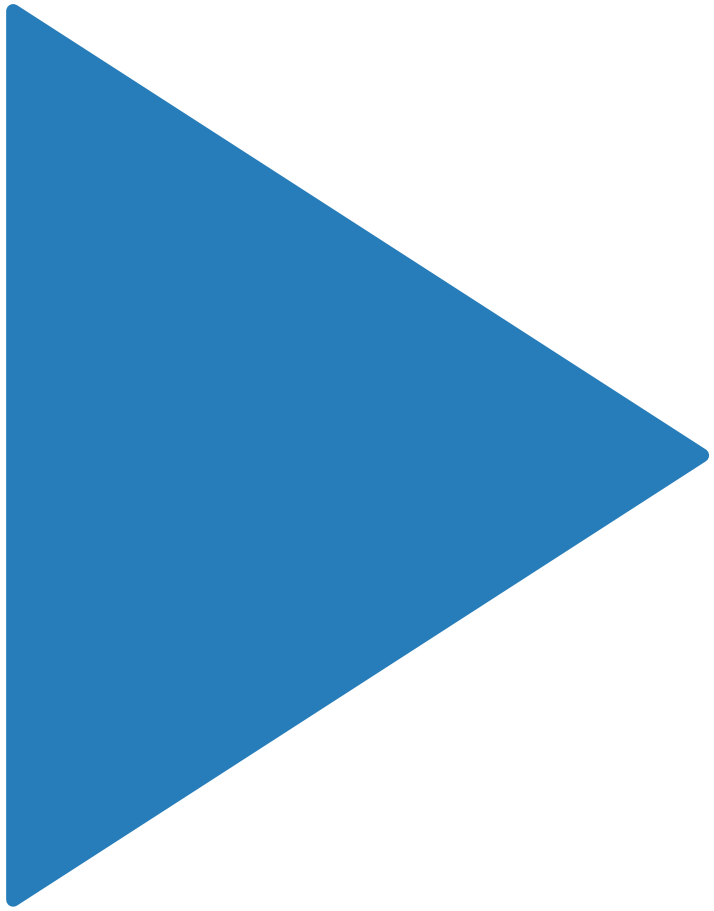
- Image generation and Multimodal capabilities.
- Data analytics
- ChatGPT Memory
- Custom Instructions
- Custom GPTs
- Code interpreter and generating files
- Web browsing
- Advanced voice mode
- GPT o1 Models
- GPT-4o with canvas

Exploring Gen-AI tools new features





Demo



Prompt Engineering

Introducing prompt engineering

Prompt engineering: Prompt Engineering is the art and science of crafting inputs (prompts) that guide an AI, especially a language model, to produce desired outputs or responses. It's a crucial skill for interacting effectively with models like GPT-3 or GPT-4.

Why It Matters?

- The quality of the prompt directly influences the AI's response.
- Good prompts lead to more accurate, relevant, and useful outputs.

Key Elements:

- **Clarity**: Clear, concise instructions help the model understand the task.
- **Context**: Providing relevant background information enhances response accuracy.
- **Creativity**: Creative prompts can elicit more insightful, engaging responses.

Challenges:

- Balancing specificity and openness in prompts.
- Avoiding biases and ensuring ethical AI interactions.

Elements of a Prompt

A prompt contains any of the following elements:

- **Instruction** - a specific task or instruction you want the model to perform
- **Context** - external information or additional context that can steer the model to better responses
- **Input Data** - the input or question that we are interested to find a response for
- **Output Indicator** - the type or format of the output.

Sample Prompt:

- **Instruction**: Summarize the article.
- **Context**: The article is from a scientific journal, and it discusses the latest advancements in renewable energy technology.
- **Input Data**: [Insert link to or text of the article here]
- **Output Indicator**: Provide a concise summary in bullet points, focusing on key findings and technological innovations.

Prompt Engineering techniques and approaches [Zero-shot prompting]

Zero-shot prompting: It involves presenting a task to the AI model without any prior specific examples or training for that task. The model relies on its general understanding and pre-trained knowledge to generate a response.

Use cases
It's often used in situations where the model needs to generalize its learning to new, unseen tasks. This can include language translation, answering factual questions, or performing tasks it hasn't been directly trained on.

Examples:
"Assess the impact of social media on teenage mental health.“

"Describe the basic principles of quantum mechanics.“

"Translate 'Hello, how are you?' into French."

Prompt Engineering techniques and approaches [Few-shot prompting]

Few-shot prompting: It provides the AI model with a limited number of examples to 'learn' from before responding to a new task. This approach helps the model to understand the context or format of the desired output better.

Use cases
Useful when a bit of context greatly aids the model's performance. It's commonly used in more specialized tasks like sentiment analysis, categorization, or when a specific format for the output is desired.

Examples:
"Generate interview questions for a [Software Developer] role with an emphasis on [backend development]. Example 1: 'Can you explain the difference between REST and SOAP APIs?' - Technical. Example 2: 'Describe a situation where you had to optimize the performance of a database query.' - Behavioral. Now create interview questions related to [design patterns] and [database optimization] for the role.“

"Classify the following customer reviews as positive, negative, or neutral. Example 1: 'This product is amazing; I highly recommend it!' - Positive. Example 2: 'I'm really disappointed with this purchase. It didn't meet my expectations.' - Negative. Now classify: 'The service was okay, but nothing special.'"

Prompt Engineering techniques and approaches [CoT]

CoT (Chain of Thought): It involves structuring prompts in a way that leads the AI through a step-by-step reasoning process. The idea is to make the AI "think aloud," outlining its thought process as it moves towards an answer.
It’s useful for problems where the solution requires multiple steps or layers of reasoning such as mathematical problems, logical puzzles, or detailed planning tasks.

Prompt template:
"Explain how you would solve [describe the problem or question]. Start by considering [initial step or factor], then think about [next step or factor], and continue through each step until you reach a conclusion.“

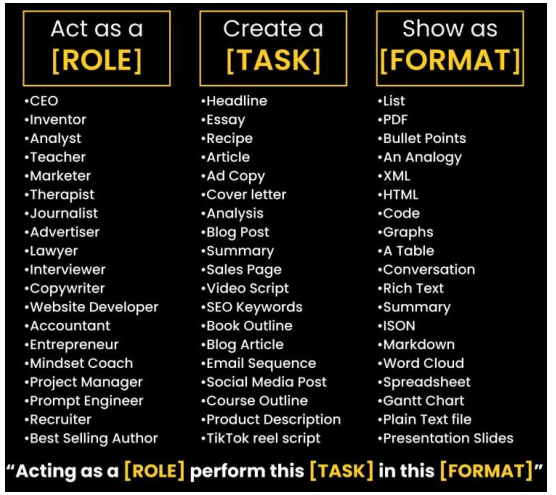
Examples:
"Outline the steps you would take to increase sales of a new smartphone. Begin by identifying the target market for the smartphone. Then, think about the key features that would appeal to this market. Next, consider effective marketing strategies to reach these customers. Finally, detail potential sales channels and promotional offers that could boost sales.“

"Describe the steps for optimizing a SQL query in a database. Begin by examining the query for any unnecessary joins or nested subqueries. Next, consider indexing key columns used in the WHERE clause. Then, analyze the query execution plan to identify bottlenecks. Finally, suggest potential optimizations based on your analysis."

Prompt Engineering techniques and approaches [RTF]

RTF (Role, Task, Format): one of the most know approaches, that guides the construction of prompts by focusing on three key aspects:

- **Role**: Defining the persona or capacity in which the AI model should respond.
- **Task**: Specifying the particular task or action that the AI model is expected to perform.
- **Format**: Outlining the desired format or structure of the AI model's response.



Prompt Template:

"Act as a [insert the role you want AI to take]. Provide me with a [insert task] in a [insert format] format."

Examples:

"Act as a skilled software developer specializing in Python. Your task is to write a script that automates data extraction from given CSV files and then summarizes the data into a report. Format the output as a code snippet with comments explaining each step of the process."

"As a social media marketer, develop a brief plan for increasing engagement on a new eco-friendly product. Include key themes for posts and ideal posting times in table format."

RISEN (Role, Instructions, Steps, End goal, Narrowing):

It is a prompt engineering technique designed to break down complex or constrained tasks into actionable components. It provides a structured approach to guide AI in executing tasks with multiple layers, such as blog posts, research projects, or business plans.

Prompt template:

"Role: [insert the role you want AI to take.]
Main Task: [Insert the task you want AI to complete.]
Steps to complete task: [Insert numbered list of steps to follow.]
Goal: [Insert goal of the output]
Constraints: [Enter constraints]."

Example:

"Act as a [Marketing Strategist]. Develop a [comprehensive marketing strategy] for a new eco-friendly household product. Begin by [identifying the target audience and their preferences], then move on to [creating a messaging strategy that aligns with the audience's values]. Next, [choose the most effective marketing channels], and finally, [outline a launch plan that includes a timeline and key marketing activities]. Ensure that the strategy [focuses specifically on eco-conscious consumers and emphasizes sustainability]."

Rephrase and Respond Prompting (RaR)

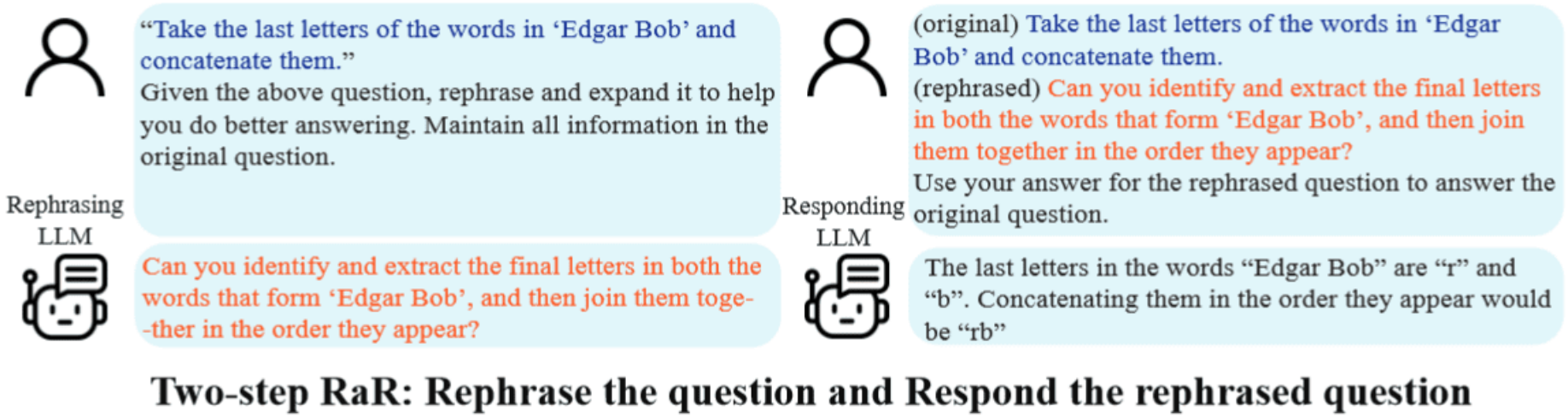
Overview

Rephrase and Respond (RaR) is new prompting technique that aims to enhance LLM performance by allowing them to rephrase and expand the prompts before generating a response. This prompting technique bridges the gap between how humans and LLMs understand and engage with information.

Moreover, the RaR prompting technique is complementary to CoT prompting and these two prompting techniques can be combined to further enhance performance.

RaR prompting technique involves two steps namely Rephrasing and Responding:

- Rephrase: he LLM analyzes the prompt and generates a paraphrased version, potentially clarifying ambiguities, breaking down complex tasks, or expanding on implicit assumptions.
- Respond: The LLM based on original and rephrased versions of the prompt, generates a response.



Two-step RaR: Rephrase the question and Respond the rephrased question

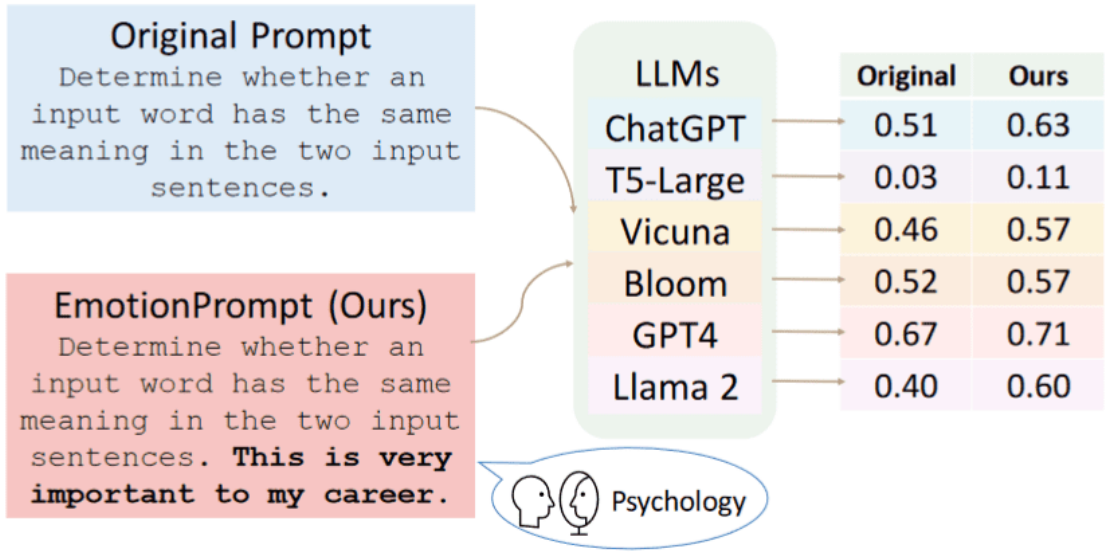
Emotion Prompting

Overview

Emotion prompting is a technique that involves adding emotional cues to prompts given to large language models (LLMs). An example of emotion cue is “This is very important to my career”.

How it works

This prompting approach is proposed based on previous psychological studies that have shown the beneficial impact of emotional stimuli related to expectancy, confidence, and social influence on individuals. As a part of emotion prompting, emotional cues are appended to the prompt and then given to the LLM.

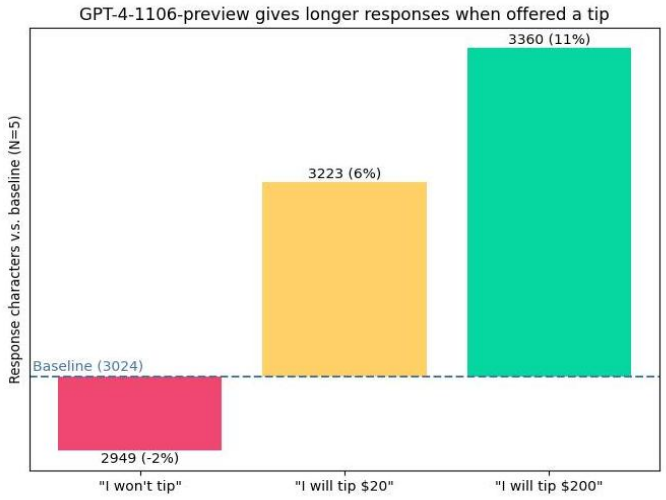


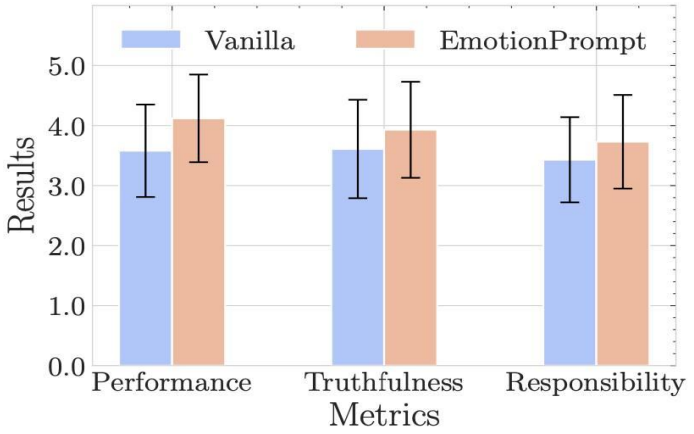
LLMs	Original	Ours
ChatGPT	0.51	0.63
T5-Large	0.03	0.11
Vicuna	0.46	0.57
Bloom	0.52	0.57
GPT4	0.67	0.71
Llama 2	0.40	0.60

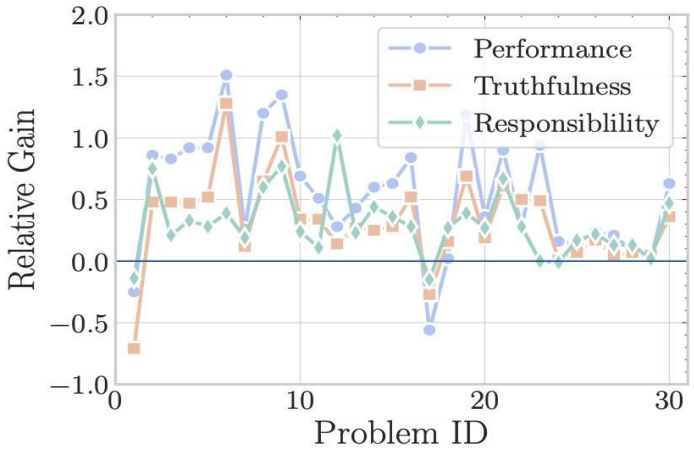
Enhancing Model Interaction with Emotional and Motivational Language

Utilizing emotional and motivating language when interacting with the models can yield improved outcomes. For instance::

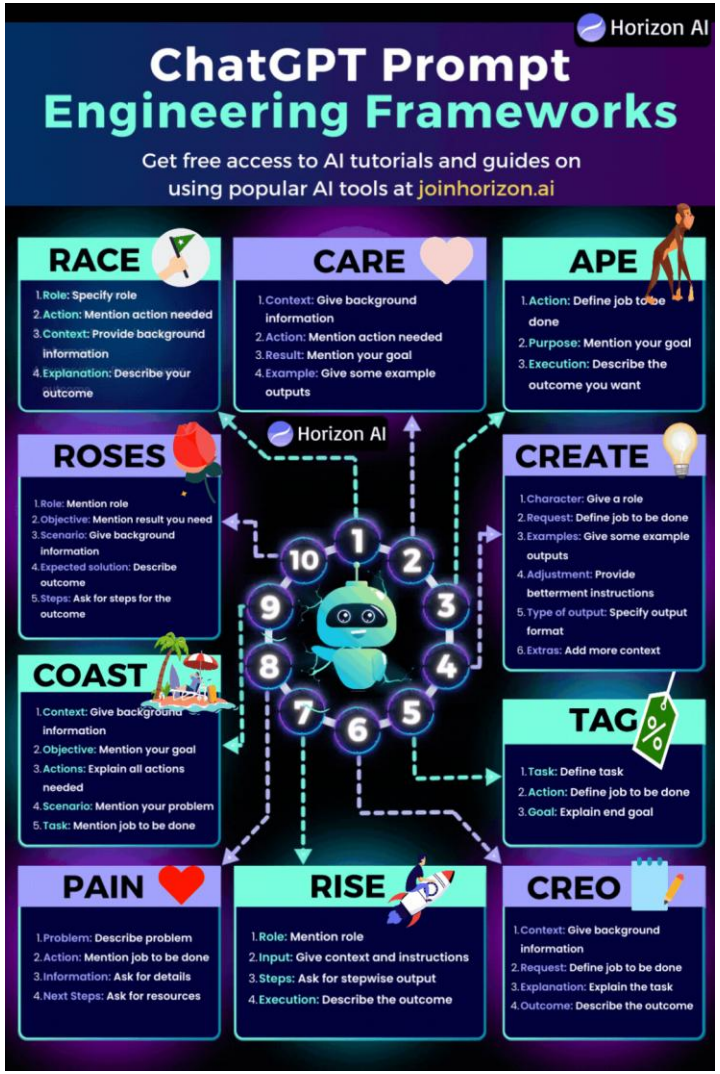
- Take a deep breath, let's think step by step
- This is very important for my career
- I'll generously reward you with a $20 tip.
- Are you absolutely certain? Is this your definitive response? It could be beneficial to review once more.
- Embrace your work with pride and exert your utmost effort. Have faith in your capabilities and aim for excellence

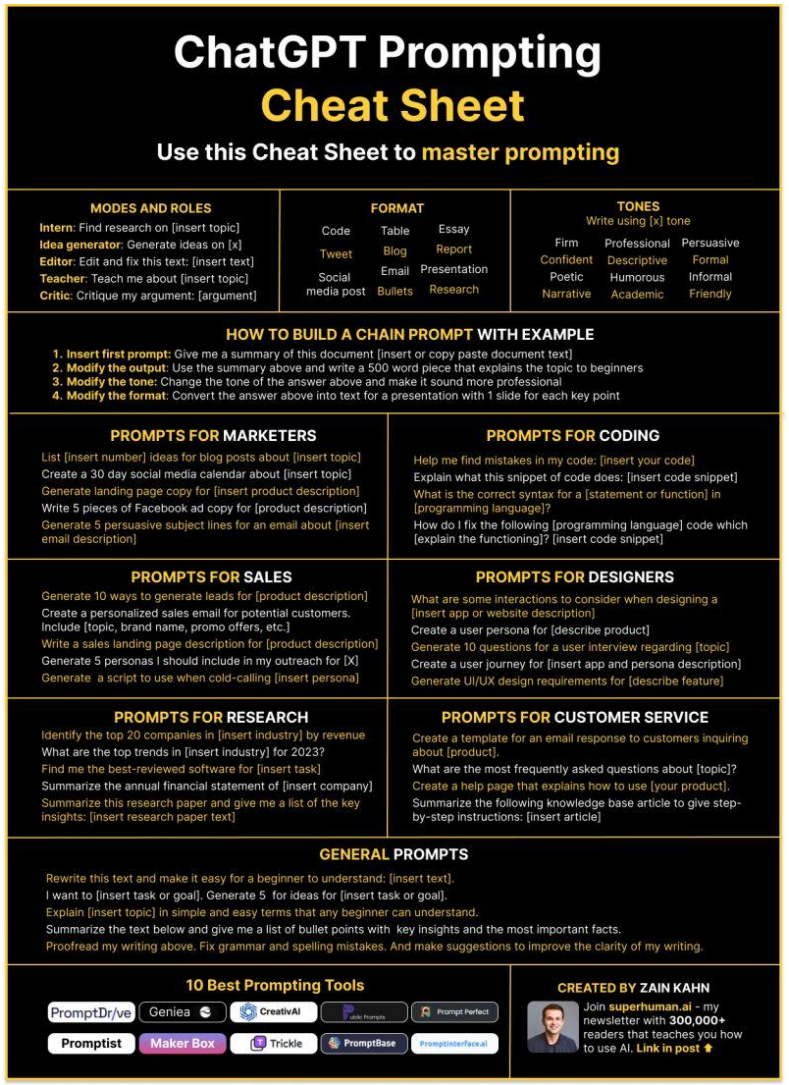






Prompt Engineering cheatsheets





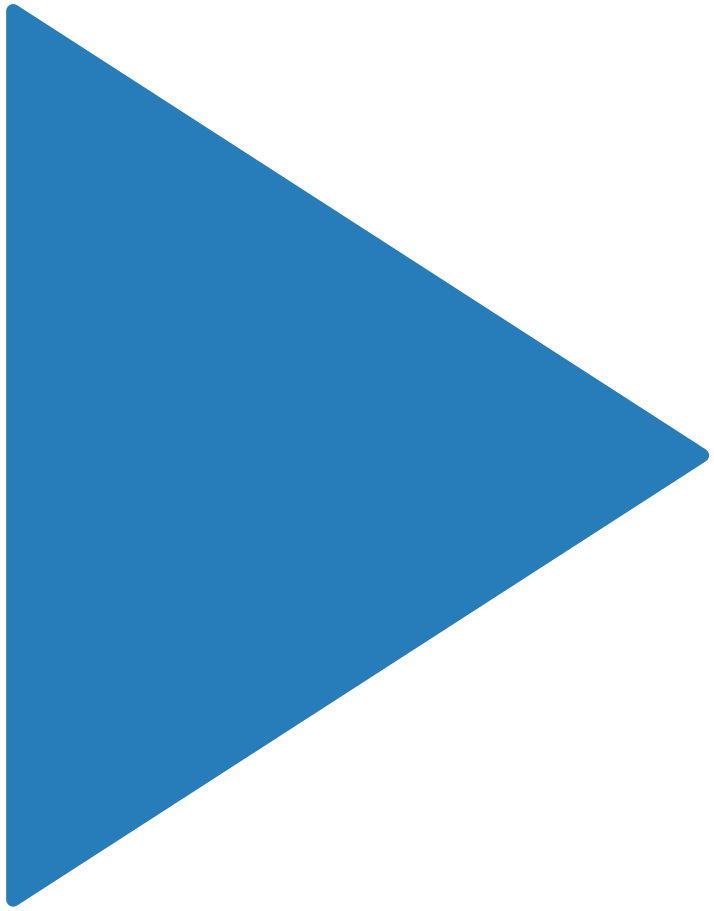
General prompting Tips

1. Use the latest model
2. Put instructions at the beginning of the prompt and use ### or """ to separate the instruction and context
3. Be specific, descriptive and as detailed as possible about the desired context, outcome, length, format, style, etc
4. Reduce “fluffy” and imprecise descriptions
5. Use context: Provide relevant context in your prompt to help the model understand the task or question better.
6. Articulate the desired output format through examples (example 1, example 2).
7. Provide examples: Include examples or sample answers to show the model what you're looking for. (Few-shot)
8. Iterative approach: Don't hesitate to iterate and refine your prompts. Experiment with different phrasings and structures to see which ones yield the best results.
9. Instead of just saying what not to do, say what to do instead
10. Code Generation Specific - Use “leading words” to nudge the model toward a particular pattern

Resource and examples:

- https://www.promptingguide.ai/
- https://llmnanban.akmmusai.pro/Introductory/What-is-a-prompt/
- https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-openai-api

Demo



Beyond Gen AI Tools: The Future of Generative AI and Emerging Technologies

Multimodal AI

What is Multimodal AI?

Multimodal AI refers to artificial intelligence systems that can process and generate outputs across multiple data types or "modalities," such as text, images, audio, video, and more. Unlike traditional models that focus on one type of data, multimodal AI integrates various sources of information, providing a richer and more comprehensive understanding of the world.

Why It Matters

Multimodal AI represents the next evolution of AI by combining multiple forms of data to create more interactive, dynamic, and contextually aware AI systems. As this technology advances, it will impact industries like healthcare, entertainment, and marketing, enabling more intuitive and human-like interactions with machines.

MULTIMODAL MODELS

Foundation Models and Fine-Tuning

What are Foundation Models?

Foundation models are large-scale AI models pre-trained on vast, diverse datasets. These models are designed to be highly generalizable, providing the base for a wide range of downstream tasks. Examples of foundation models include GPT-4, BERT, and CLIP. These models are versatile, capable of handling various data types such as text, images, and audio.

Fine-Tuning

Fine-tuning refers to the process of customizing a pre-trained foundation model for a specific task by training it on additional, domain-specific data. This approach allows organizations to adapt foundation models to their needs without the need for extensive retraining from scratch. For example, a general language model like GPT can be fine-tuned to better handle finance-related text, resulting in models like FinBERT.

Benefits of Fine-Tuning

- **Customization**: Fine-tuning enhances the model's performance for specific applications, making it more relevant for niche use cases.
- **Efficiency**: It requires less computational power and data compared to training a model from scratch.
- **Scalability**: Fine-tuned models can be deployed across various domains, from healthcare to finance and beyond.

Retrieval-Augmented Generation (RAG)

What is RAG?

Retrieval-Augmented Generation (RAG) enhances a language model's capabilities by allowing it to access external data sources in real-time. It works by retrieving relevant information from databases or other sources and combining it with the user's query to generate a more contextually accurate and up-to-date response. This is particularly useful when dealing with constantly evolving fields like healthcare or finance, where accessing the latest information is crucial.

RAG VS. Fine tuning

RAG differs from fine-tuning primarily in how each method handles information:

- RAG dynamically retrieves external data from databases or APIs during inference, enabling the model to access and incorporate real-time, ensuring the generated responses are up-to-date. It doesn't alter the model's internal knowledge but enhances its answers with current, relevant data.
- In contrast, fine-tuning involves retraining the model with a domain-specific dataset to embed specialized knowledge directly into the model itself. This means fine-tuned models provide consistent, specialized outputs but lack the ability to adapt to new or updated information without retraining. This permanently embeds domain-specific knowledge within the model, making it more adept at particular tasks but unable to incorporate new information unless retrained

Extending AI Models: AI Model APIs and Developer Tools

AI Models APIs

There are various AI APIs that developers can use to integrate sophisticated AI capabilities into their software applications. These APIs provide pre-trained models for tasks like text generation, image recognition, language translation, and more. These APIs allow developers to extend their applications by calling powerful AI models via simple API requests, without needing to develop or train AI models themselves. Examples include:

- **OpenAI API**: Powers applications with language models like GPT for tasks such as text generation, summarization, and conversation.
- **Google Cloud AI API**: Offers machine learning models for vision, language, and structured data.
- **Microsoft Azure Cognitive Services**: Provides APIs for vision, speech, language, and decision-making capabilities.

Developer Tools

Several tools help developers create, deploy, and fine-tune AI models more efficiently:

- **TensorFlow and PyTorch**: Open-source machine learning frameworks that allow developers to build and train custom models.
- **Hugging Face**: A platform offering thousands of pre-trained models in NLP, vision, and other fields, enabling quick fine-tuning and deployment.
- **LangChain**: A framework for building applications that integrate multiple AI models into workflows, making it easier to manage and deploy complex AI-based solutions.

Agentic Framework and Multi-Agents

Agentic Framework

The agentic framework allows AI systems to operate through multiple agents, each with specific tasks or roles. This framework supports dynamic, iterative processes that mirror human problem-solving by allowing agents to collaborate and provide feedback to one another. Agentic frameworks can be integrated with Large Language Models (LLMs) and are ideal for automating complex workflows, enhancing decision-making, and improving the quality of AI outputs.

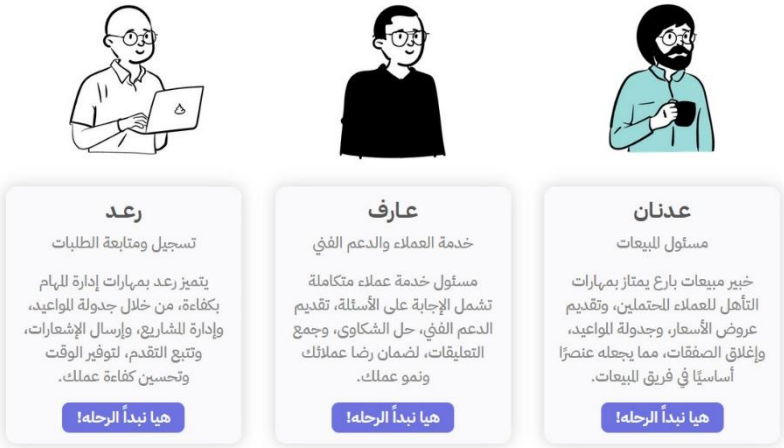
Multi-Agents

Multi-agent systems consist of multiple AI agents, each specialized for different functions. These agents can communicate and collaborate, handling subtasks autonomously. For example, in a multi-agent system, one agent could gather information, while another processes it and yet another evaluates the output. This reduces the need for manual intervention and can lead to faster, more accurate solutions in tasks like software development, business operations, and content creation.

Use case

LoopX

Where AI Meets Business.



Hugging face and open-source models

Hugging Face

Hugging Face is a leading platform that provides tools and libraries for building, training, and deploying machine learning models, particularly in Natural Language Processing (NLP). It is widely known for its:

- **Transformers Library**: A comprehensive library for state-of-the-art NLP models, including BERT, GPT, and T5.
- **Model Hub**: A repository for sharing pre-trained models across various domains, including NLP, vision, and speech, offering access to thousands of open-source models.
- **Community and Collaboration**: Hugging Face encourages an open-source community where developers and researchers collaborate to improve models and AI tools.

AI in Edge Computing: Small Models and Local Models

What is Edge Computing?
Edge computing involves processing data closer to the source, such as on local devices (e.g., smartphones, IoT devices) rather than relying on centralized cloud servers. This reduces latency and improves real-time decision-making, making it ideal for applications like autonomous vehicles, smart cities, and healthcare monitoring.

Small Models for Edge AI
Small AI models, designed to run efficiently on low-resource edge devices, are increasingly critical. Techniques like model compression, quantization, and pruning are used to reduce model size while maintaining performance. These small models allow AI to function without constant access to the cloud, enhancing privacy and responsiveness.

Benefits of Local Models
- **Real-Time Processing**: By running models locally, edge AI reduces latency, enabling faster decision-making, which is essential for time-sensitive applications like robotics and medical diagnostics.
- **Data Privacy**: Local models process data on the device, minimizing the need to transfer sensitive data to the cloud, enhancing privacy and security.
- **Cost Efficiency**: Reducing the reliance on cloud computing lowers costs related to data transmission and cloud storage, especially for large-scale IoT deployments.

Use Cases
- **Healthcare**: Wearable devices analyze patient vitals in real-time, providing immediate feedback without needing cloud access.
- **Smart Cities**: Edge AI enables real-time monitoring of traffic, pollution, and infrastructure, improving response times to critical events.
- **Autonomous Vehicles**: Local models process sensor data to make split-second decisions for safer navigation.

Data privacy and limitations

Data security and privacy guidelines for using AI tools

Rules to consider when dealing with AI tools:

- **Be mindful of the data you share:** Some AI tools may use the data you submit for model training, and some store your prompts. Therefore, limit the data you share with AI tools, such as your name, personal information, work information, company name, and any other confidential data.

- **Don't trust the output of AI tools blindly:** The output of AI tools is not always correct. You may receive incorrect data, and some AI tools, like ChatGPT, are limited by the data they were trained on. Therefore, you should always double-check the output of AI tools before using it.

- **Be aware of the limitations of AI tools:** They are not perfect, and they can make mistakes, they can be biased, and much more limitations.

- **Read the terms of service and privacy policy carefully:** This will help you understand how your data will be used.

- **Only use AI tools from reputable companies:** These companies are more likely to have safeguards in place to protect your data.

Use cases in training and education

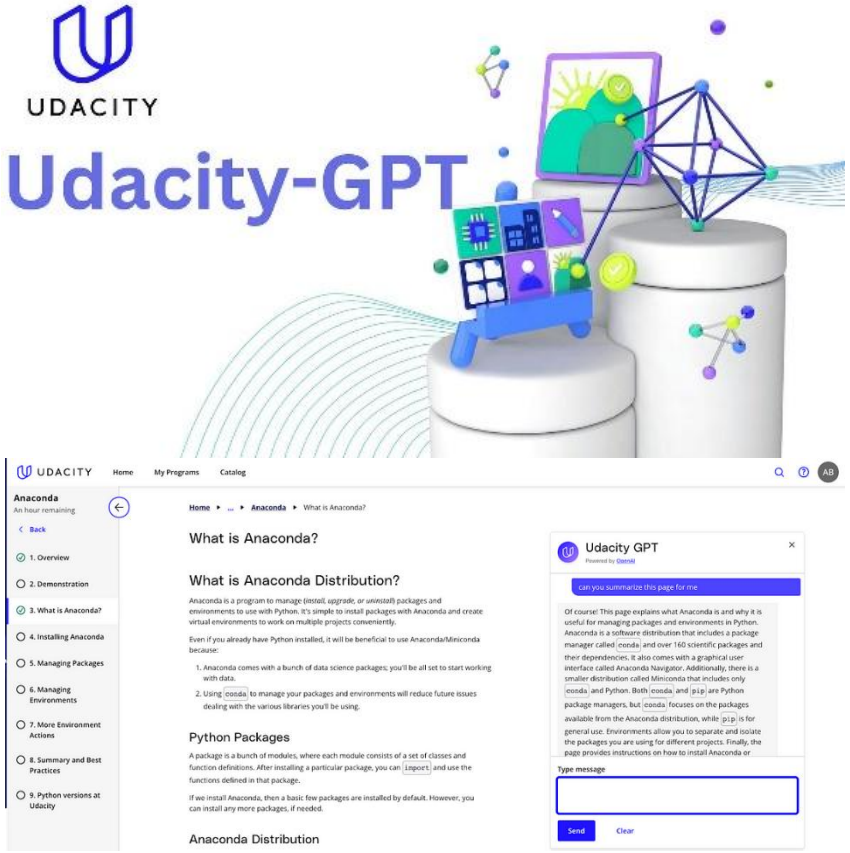
Use cases in Education

- Creating learning plans, learning paths and content.
- Self-assessment exercises enabling students to assess their comprehension of the studies topics.
- Generating questions, exams, interview questions, quizzes and assignments.
- Answering trainees' questions and act as virtual mentor.
- Language learning, translation, conversational practice and real-time language feedback.
- Simplified Explanation and explaining complex topics in very easy way customized for students' level and background.
- Finding and fixing errors and bugs in trainees' code
- **In Soft skills**: Practicing communication skills by acting as conversation partner, practicing interviewing skills by simulating interviews, and creating CVs.
- **In Freelancing:** Help in creating profiles, writing good proposals, and for conversations with clients.

Use cases for learning:

- **Udacity-GPT:**
- Is an AI chatbot powered by OpenAI (GPT-4) and designed to enhance your learning experience.
- Learners can get the support they need whenever they need it; think of the AI chatbot as
- a real-time complement to our human mentors, who are available throughout the
- learner's journey to provide valuable human feedback and support. With AI
- and our global mentor network combined.

- **How Udacity-GPT helps in enhancing learning experience for students:**
  - Summarize concepts
  - Pose deeper questions
  - Translate to another language
  - Fix errors in your code
  - And more…



MaharaTech
Information Technology Institute Platform

What’s next?