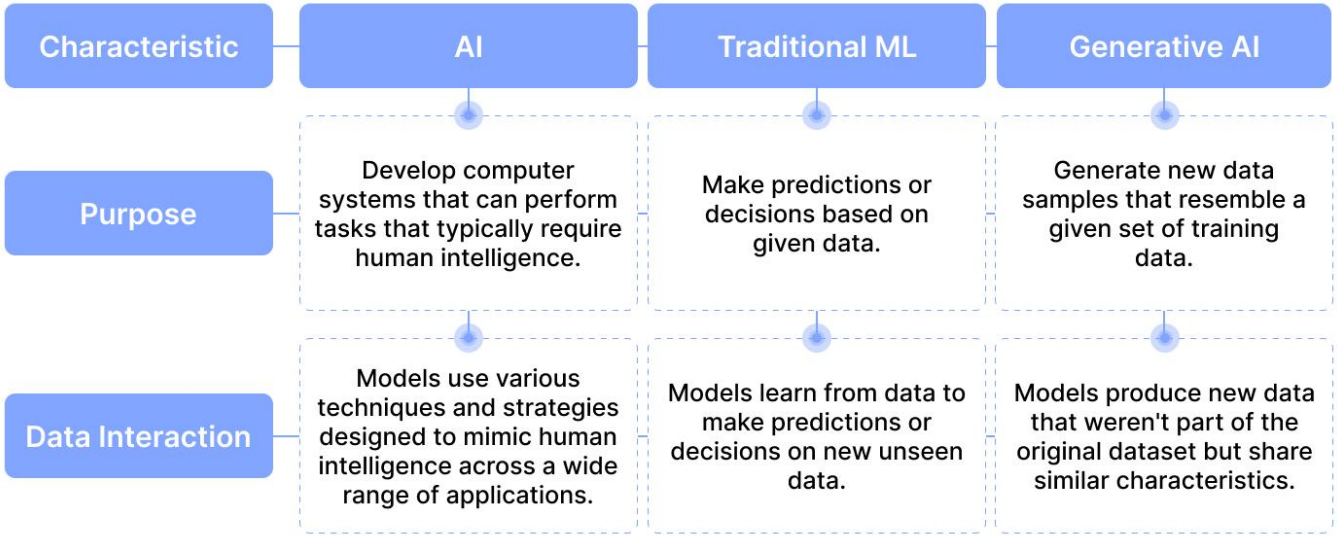
Generative AI

01

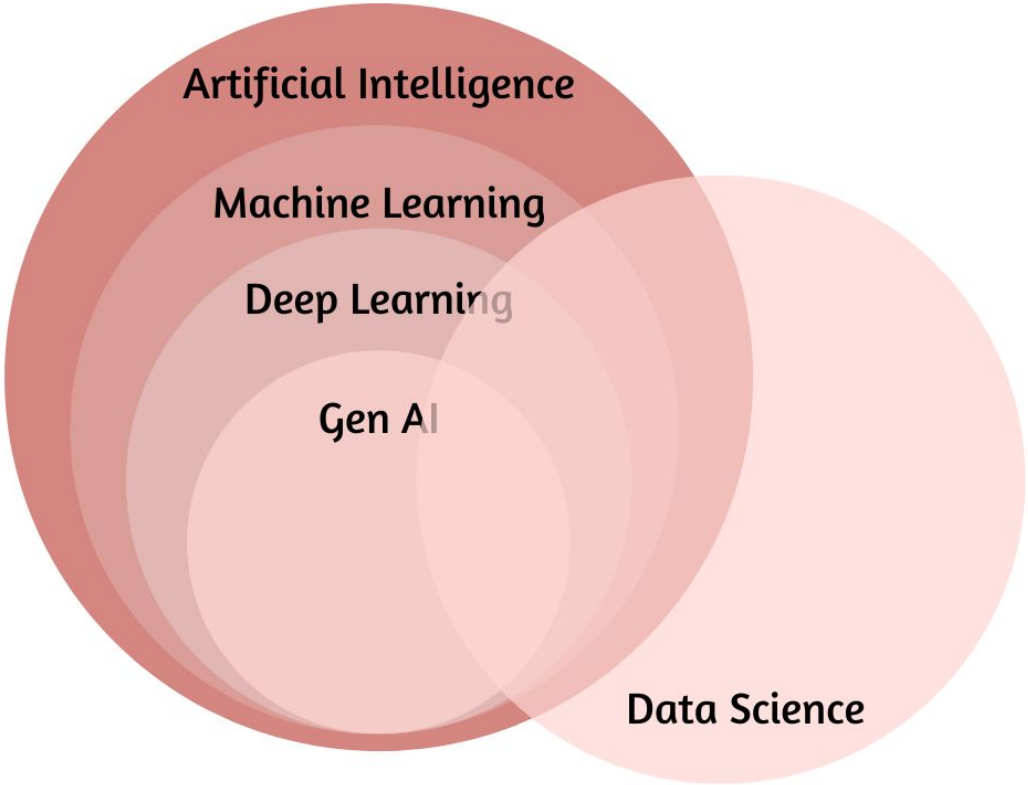
Overview

- Generative AI (genAI) is a broad label describing any type of artificial intelligence (AI) that can produce new text, images, video, or audio clips. Technically, this type of AI learns patterns from training data and generates new, unique outputs with the same statistical properties.
- Generative AI models use prompts to guide content generation and use transfer learning to become more proficient.

Artificial Intelligence vs. Traditional Machine Learning, Generative AI



Technopedia


Artificial Intelligence
Machine Learning
Deep Learning
Gen AI
Data Science

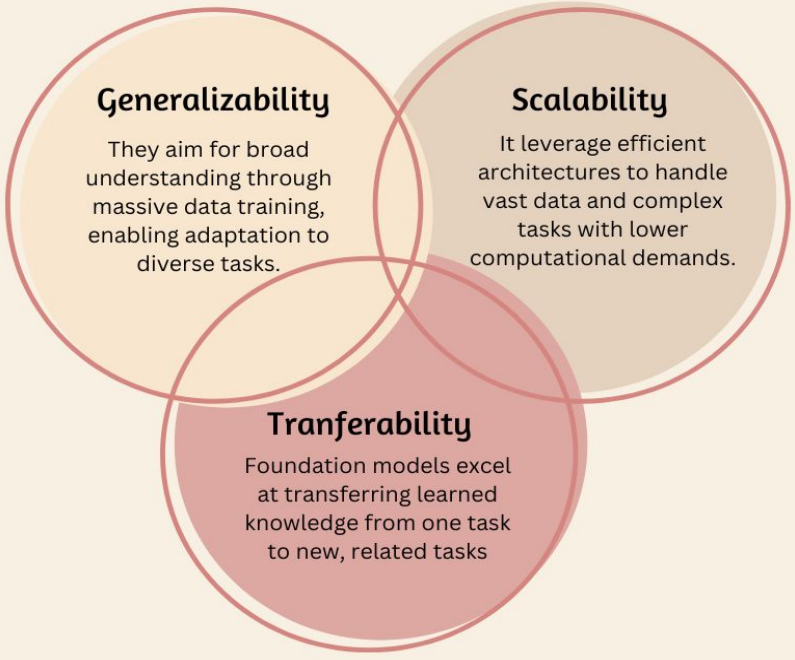
How GenAI work?

- Generative AI, at its core, is about predicting the next piece of data in a sequence, whether that's the next word in a sentence or the next pixel in an image.
- GenAI is actually built on "Large Language Models" (LLM not AI), and these LLMs are based on the "Generative Pre-trained Transformer" architecture (or GPT—as invented by Google). These models learn patterns from a massive amount of text data (ChatGPT 3.5 was trained on 175B parameters, whilst ChatGPT-4 is trained on 1 Trillion parameters.)
- LLMs use these patterns (and not logic) to generate responses (they are basically *very* powerful autocorrect). Unlike other computer systems that are particularly good at math, LLMs are subject to "hallucinations," where they may generate seemingly meaningful responses, that are not otherwise correct.

Foundation Models

Foundation models are AI models designed to produce a wide and general variety of outputs. They are capable of a range of possible tasks and applications, such as text, image or audio generation. They can be standalone systems or can be used as a ‘base’ for many other applications.

Characteristics of Foundation Models



Types of GenAI models

- Text to Text
  - BERT
  - T5
- Text to Image
  - DALL-E 2
  - Stable Diffusion
- Image to text
  - Flamingo
- Image to 3D
  - 3D LLM
- Image or video to 3D
  - 3D LLM

Types of GenAI models

- Text to Audio
  - Tacotron
- Text to code
  - Codex
  - Alphacode
- Image to Science
  - Galactica
- Text to video
  - MovieNet
- Audio to text
  - Whisper

Open vs Closed Source LLM

Close source vs Open Source LLM

- **Accessibility**
  - Not publicly available. Access and usage are restricted by developers who own them
- **Customization**
  - Limited customization options. Users typically rely on pre-defined parameters or APIs provided by the developers.
- **Cost & Availability**
  - Often require licensing fees or pay-per-use models for access.
- **Example**
  - GPT-3 (OpenAI), Jurassic-1 Jumbo (AI21 Labs)

- **Accessibility**
  - The source code is publicly accessible. Anyone can view, modify, and distribute the code.
- **Customization**
  - Greater flexibility for customization. Developers can modify the code to fit specific needs and integrate them into complex systems.
- **Cost & Availability**
  - Freely available to use and modify, with minimal to no cost
- **Example**
  - BLOOM (Hugging Face), Megatron-Turing NLG (NVIDIA)

Example of Closed source LLM

GPT 3.5

- **Developed by:** OpenAI
- **Release Date:** Nov 2022
- **License:** Closed-Source
- **Supported Natural Languages:** Mainly English, Multilingual Possible
- **Supported Programming Languages:** Python, JavaScript, Java, C/C++,HTML/CSS, R,SQL, Swift, Ruby, PHP
- **Number of Model Parameters:** 175 billion parameters
- **Number of Tokens in Dataset:** dataset is not publicly disclosed
- **Famous Applications:** ChatGPT, AI Writer by Jasper, Jarvis

Gemini

- **Developed by:** Google Deepmind
- **Release Date:** Dec 2023
- **License:** Closed-Source
- **Supported Natural Languages:** Mainly English
- **Supported Programming Languages:** C/C++,C#, Bash, Dart, Go, GoogleSQL, Java, JavaScript, Kotlin, Lua, MatLab, PHP, Python, R, Ruby, Rust, Scala, SQL, Swift, TypeScript, YAML
- **Number of Model Parameters:** At Least 1.8 billion parameters
- **Number of Tokens in Dataset:** dataset is not publicly disclosed
- **Famous Applications:**

Claude 3

- **Developed by:** Anthropic
- **Release Date:** March 2024
- **License:** Apache 2.0
- **Supported Natural Languages:** Mainly English, French, Japanese, Spanish
- **Supported Programming Languages:** Python, JavaScript, Java, C#, C++, GO, Swift, Ruby, PHP, Kotlin
- **Number of Model Parameters:** 137 billion parameters
- **Number of Tokens in Dataset:** dataset is not publicly disclosed
- **Famous Applications:**

Example of Open Source LLM

BERT

- **Developed by:** Google
- **Release Date:** Oct 2018
- **License:** Apache License 2.0
- **Supported Natural Languages:** Multilingual
- **Supported Programming Languages:** NO
- **Number of Model Parameters:** 110 million parameters
- **Number of Tokens in Dataset:**
- **Famous Applications:**

LLaMA 2

- **Developed by:** Meta
- **Release Date:** July 2023
- **License: Llama 2 Community License**
- **Supported Natural Languages:** Multilingual
- **Supported Programming Languages:** Python, C++, Java, PHP, C#, Bash
- **Number of Model Parameters:** 7 billion parameters
- **Number of Tokens in Dataset:**
- **Famous Applications:**

Falcon

- **Developed by:** TII
- **Release Date:** May 2023
- **License:** Apache License 2.0
- **Supported Natural Languages:** English, German, Spanish, French, Italian, Portuguese,Polish, Dutch, Romanian, Czech, Swedish
- **Supported Programming Languages:** NO
- **Number of Model Parameters:** 7 billion parameters
- **Number of Tokens in Dataset:**
- **Famous Applications:**

Application of LLM

Application

- **ChatGPT:** Used for text generation, text completion, text classification, text summarization
- **Jasper.ai:** is an AI writing assistant that helps you create and improve content faster by using AI-powered generation and editing tools.
- **ChatPDF:** is an AI-powered PDF assistant that lets you chat with your PDFs to ask questions, summarize content, and navigate documents easily.
- **Google Bard:** help you write, translate, code, and explore information through conversation.
- **Github Copilot:** is your AI pair programmer that suggests code completions and functionalities as you type.
- **10web.io:** is an all-in-one website builder platform offering AI-powered tools for building and managing websites with ease.

Application

- **Wordtune:** is an AI writing assistant that helps you improve your writing by suggesting rephrase sentences, adjusting tone, and even generating creative text formats.
- **Soundraw**: It lets you create royalty-free music with AI, in just a few clicks.
- **Midjourney:** is an AI that generates images from descriptions you give it in natural language.
- **Synthesia AI:** It creates studio-quality videos with AI avatars and voices, so you can make videos without filming yourself.
- **Starry Tars:** Generates avatars

Techniques to build LLM Application

1: Prompt Engineering

- Prompt engineering refers to the strategic design and construction of input prompts to guide Language Models (LMs) towards desired outputs.
- It aims to influence the behavior and outputs of LMs by providing specific context and instructions through well-crafted prompts.
- Components of Prompt Engineering
    - **Contextual Prompts**: Include relevant information or context that helps LMs understand the desired task or domain.
    - **Control Codes:** Special tokens or instructions embedded within prompts to direct LMs on aspects like style, tone, or content generation.
    - **Evaluation Metrics:** Criteria used to assess the quality and effectiveness of prompt-engineered outputs.

2: Retrieval Augmented Generation (RAG)

- RAG combines retrieval-based methods with generation models to produce more accurate and contextually relevant outputs.
- Enhance the performance of LMs by retrieving relevant information from external sources and integrating it into the generation process.
- Components of Retrieval-Augmented Generation
  - **Retrieval Module**: Searches and retrieves relevant documents or data from a predefined corpus.
  - **Generation Module:** Uses the retrieved information to generate coherent and contextually accurate text.
  - **Integration Mechanism:** Seamlessly combines retrieved data with generative capabilities to produce high-quality outputs.

3: Fine Tuning LLMs

- Fine-tuning involves adapting a pre-trained LLM to a specific task or domain by further training it on a smaller, task-specific dataset.
- Enhance the model's performance on particular tasks, making it more accurate and relevant to specific applications.
- Process of Fine-Tuning LLMs
    1. **Pre-Training:** Start with a pre-trained LLM that has been trained on a large and diverse corpus.
    2. **Data Preparation:** Collect and preprocess a task-specific dataset for fine-tuning.
    3. **Training:** Train the pre-trained LLM on the task-specific dataset, adjusting model weights to improve performance.
    4. **Evaluation:** Evaluate the fine-tuned model on a validation set to ensure it meets the desired performance criteria.

4: Training LLM from scratch

- Training LLMs from scratch involves building a language model starting with an untrained neural network, using a large and diverse corpus of text.
- Develop highly customized and powerful language models tailored to specific requirements without relying on pre-existing models.
- Challenges
  - **Data Requirements**: Need for vast amounts of high-quality data to achieve good performance.
  - **Computational Resources:** Significant computational power and time required for training large models.
  - **Expertise**: Requires substantial expertise in machine learning, NLP, and neural network architecture design.

Beneficial LLM App

Offensive

Ugly

Prompt Injection Attack on LLM

- Prompt injection is a type of adversarial attack where malicious prompts are strategically injected into input data to manipulate the behavior and outputs of LLMs.
- **Purpose:** Exploit vulnerabilities in LLMs to generate biased, misleading, or harmful responses.
- Techniques for Prompt Injection
  - **Textual Injection:** Directly injecting malicious prompts into input text to influence model outputs.
  - **Contextual Manipulation:** Leveraging contextual cues or information to guide the model towards generating desired outputs.
  - **Adaptive Injection**: Dynamically adjusting injected prompts based on model responses to maximize attack effectiveness.

Jailbreaking Attack on LLM

- Jailbreaking is a type of security attack where malicious actors attempt to bypass or exploit security measures in LLMs to gain unauthorized access or control.
- **Purpose**: Gain unauthorized access to sensitive information, manipulate model behavior, or extract proprietary data.
- Techniques for Jailbreaking Attacks
  - **Parameter Tampering:** Manipulating model parameters to influence outputs or introduce vulnerabilities.
  - **Adversarial Inputs:** Crafting inputs designed to exploit weaknesses in the model's decision-making process.
  - **Exploiting Runtime Environments:** Leveraging vulnerabilities in runtime environments or APIs used by the LLM.

Exploiting Hallucinations in LLM

- Hallucinations occur when LLMs generate plausible-sounding but incorrect or nonsensical outputs.
- **Cause:** Can arise due to model limitations, insufficient training data, or inherent complexities in natural languages.
- Examples of Hallucinations in LLMs
  - **Incorrect Factual Information** Example: LLMs generating false historical dates or events.
  - **Fabricated References** Example: LLMs inventing non-existent books, articles, or authors.
  - **Illogical Conclusions** Example: LLMs providing answers that defy logic or context.

Reference

- https://www.techopedia.com/definition/34633/generative-ai
- https://yellow.ai/blog/types-of-generative-ai/
- https://www.analyticsvidhya.com/blog/2023/03/an-introduction-to-large-language-models-llms/
- https://www.databricks.com/resources/webinar/build-your-own-large-language-model-dolly
- https://www.analyticsvidhya.com/blog/2023/05/foundation-models/