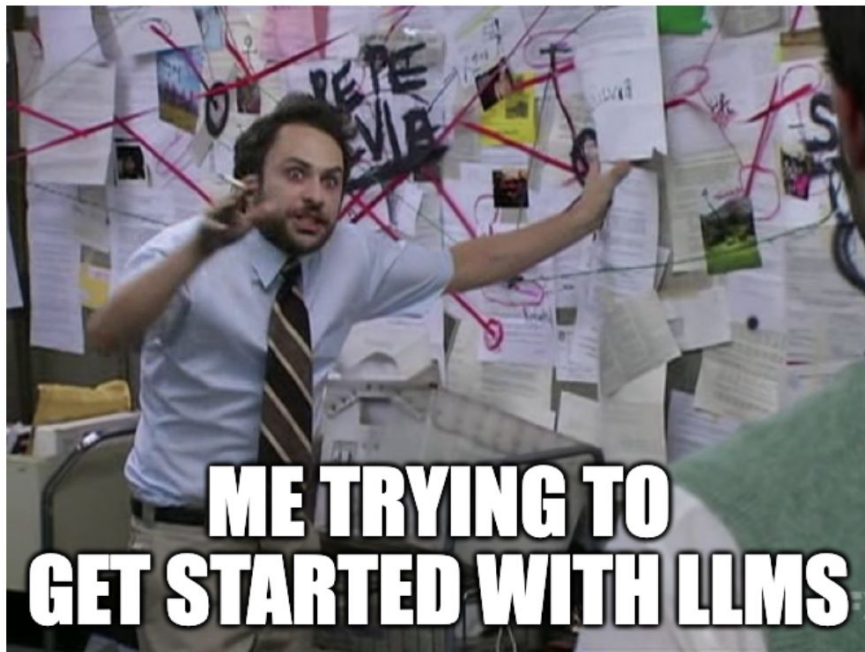


GenAI

October 2024

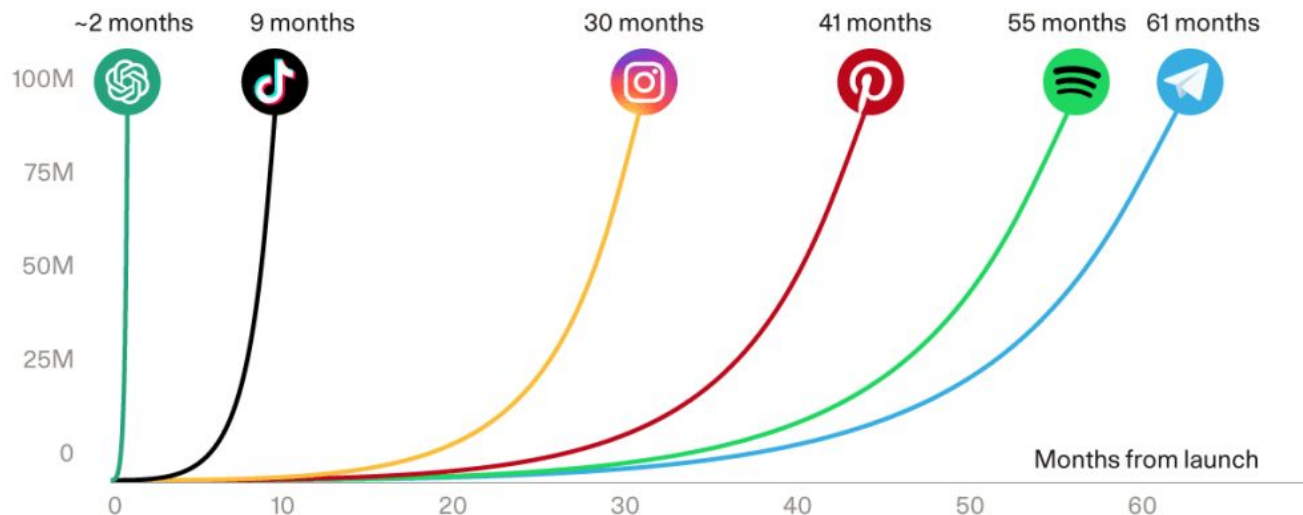


What to expect today – Intro to Gen AI



Why Now? The Incredible Growth of ChatGPT

Path to 100 Million Users (stylized)



Harvard study finds consulting BCG's staff using ChatGPT outperform on every measure

Conducted an experiment in using GPT-4 to augment consulting work

A randomized control trial (experiment) involving hundreds of consultants found that those with access to GPT-4 outperformed those without in **eighteen different tasks** for a fictional shoe company, **across metrics like time, task completion, and output quality**.

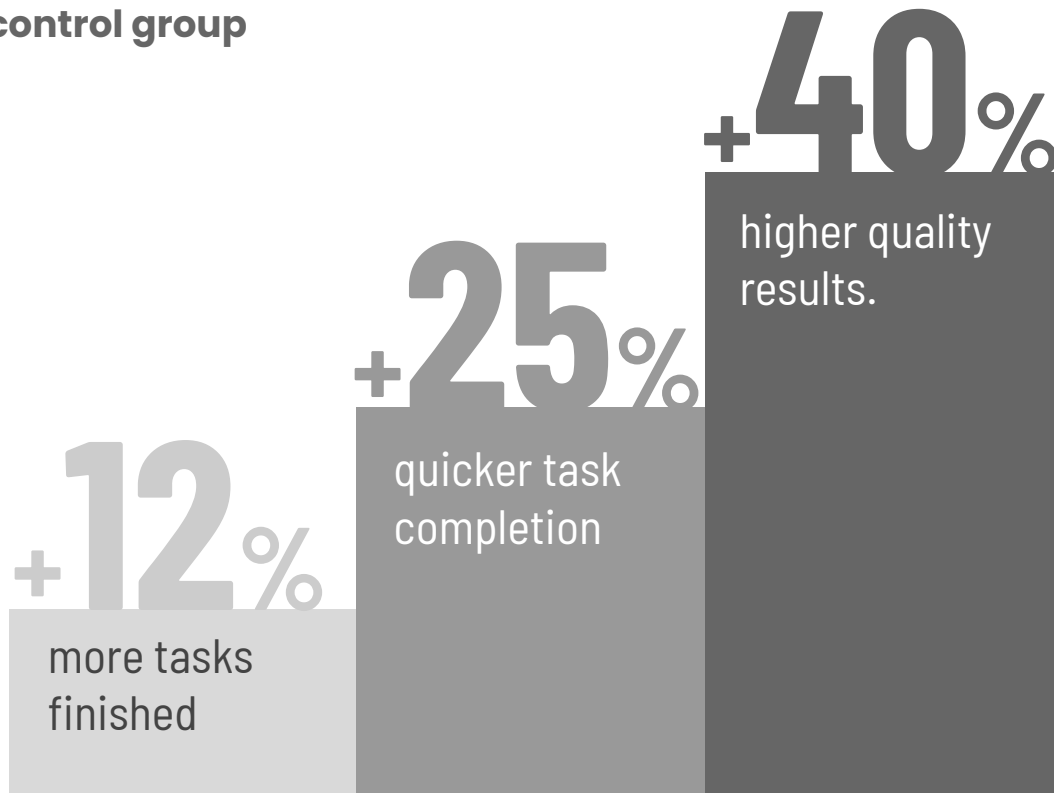
- GPT-4 empowered consultants were more efficient and produced higher-quality work in **creative, analytical, and writing tasks**.
- Both AI and human graders rated GPT assisted work as higher quality.

Think About It: Performance Boost

Guess the performance boost (%) for consultants
when working with GPT-4

Consultants using GPT-4 show higher quality results

Versus the control group



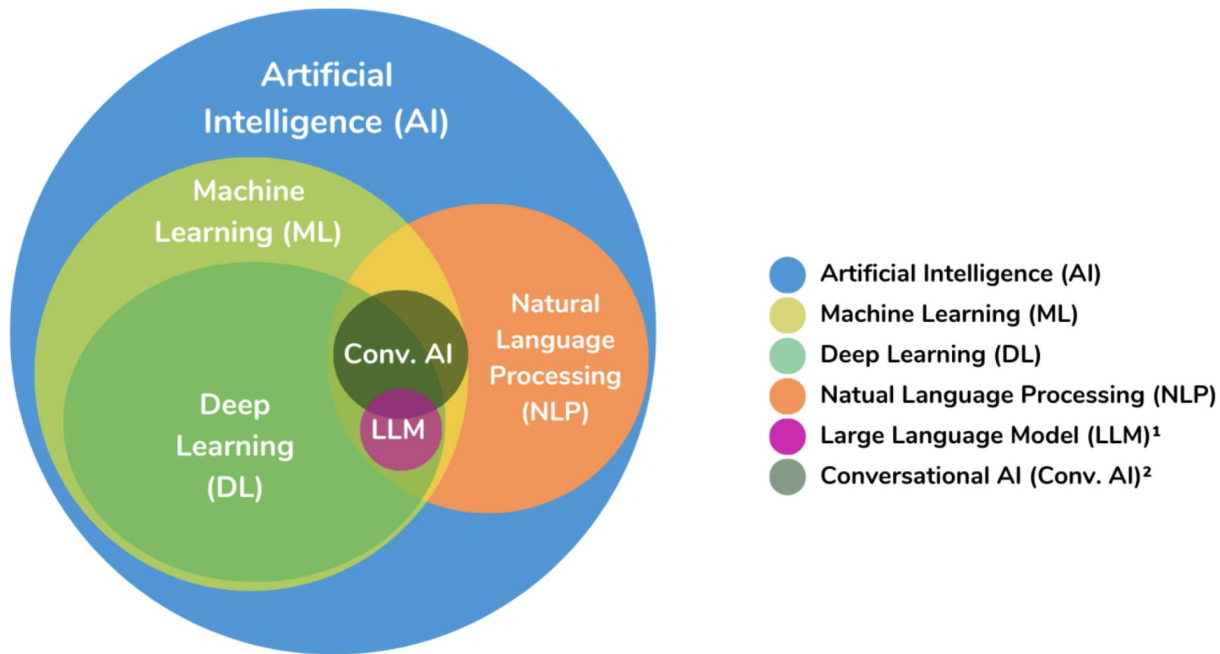
More details and
research paper - [here](#)

AI WON'T REPLACE HUMANS, BUT HUMANS WITH AI WILL REPLACE HUMANS WITHOUT AI

Karim Lakhani,
Professor at Harvard Business School

LLM

AI Landscape



LLM: Large Language Models

Large Language Models (LLMs) are a type of **transformer** model that is trained on vast amounts of text data to understand and generate human-like language. They are called "large" because they typically have billions or even trillions of parameters, which allows them to capture intricate patterns and nuances in language.

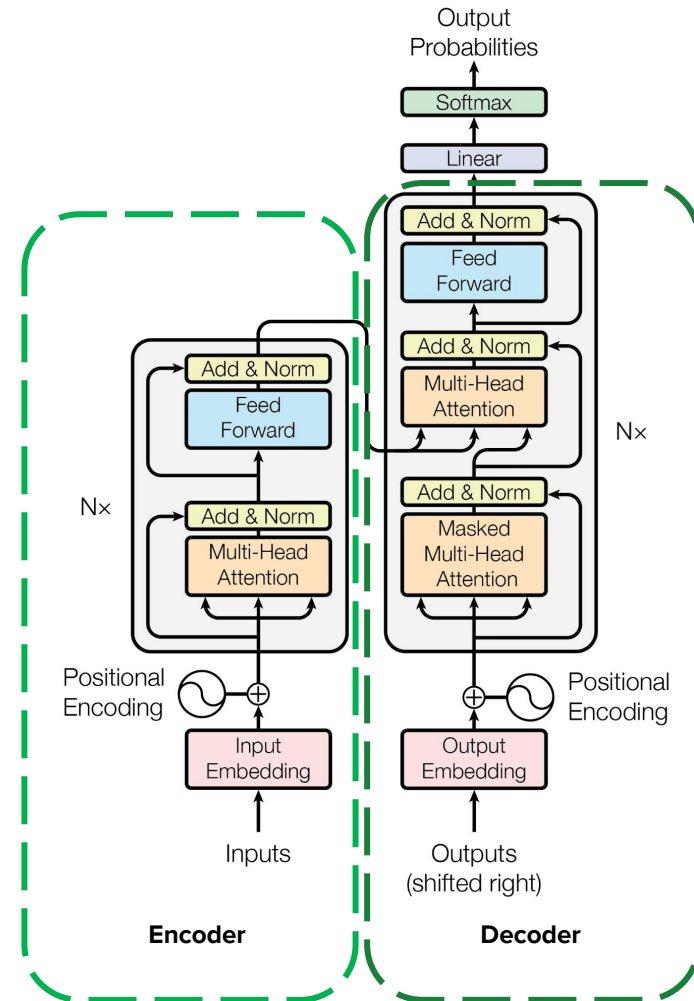
LLM Model Variants come in different sizes and architectures, each with its own strengths and trade-offs

Key Concept: Transformers

A Transformer is a deep learning model architecture introduced in the paper “Attention is All You Need” by Vaswani et al. (2017). It relies entirely on self-attention mechanisms to process sequences of data.

Architecture: Comprises an encoder-decoder structure

- **Self-Attention Mechanism:** Allows the model to weigh the importance of different words in a sentence when encoding a single word.
- **Positional Encoding:** Adds information about the position of each word in the sequence, since Transformers do not inherently consider word order.
- **Multi-Head Attention:** Multiple attention mechanisms work in parallel, allowing the model to focus on different parts of the sentence simultaneously.
- **Feed-Forward Neural Networks:** Layers that process the output of the attention mechanisms, adding non-linearity and enhancing the model's capacity.
- **Layer Normalization:** Applied after each sub-layer (attention and feed-forward) to stabilize and speed up training.
- **Residual Connections:** Shortcuts around sub-layers to help with gradient flow during training



Key Concepts: Model size

Models

- Domain-specific models: Fine-tuned on data from specific domains like finance, healthcare, etc.
- Task-specific models: Optimized for tasks like summarization, translation, code generation, etc.
- Multilingual models: Trained on data from multiple languages

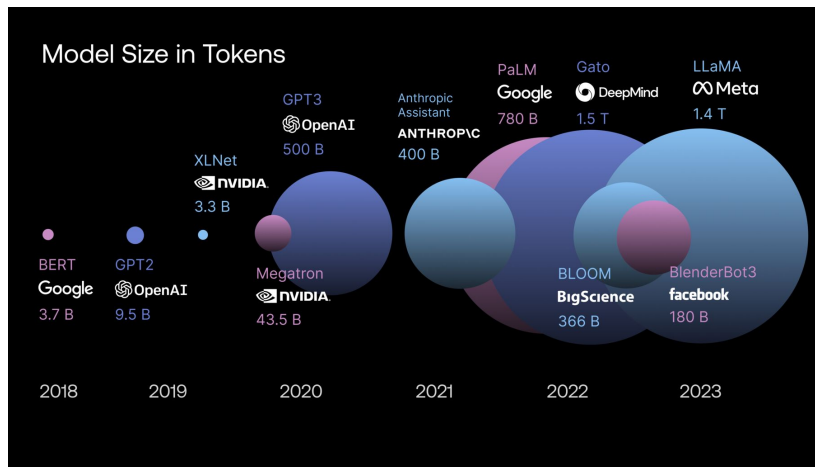
Model Size

Larger models have better performance but higher compute cost, so it comes at a tradeoff to run bigger models.

For reference: GPT 4 has over 1.7T

Model Selection Best Practices

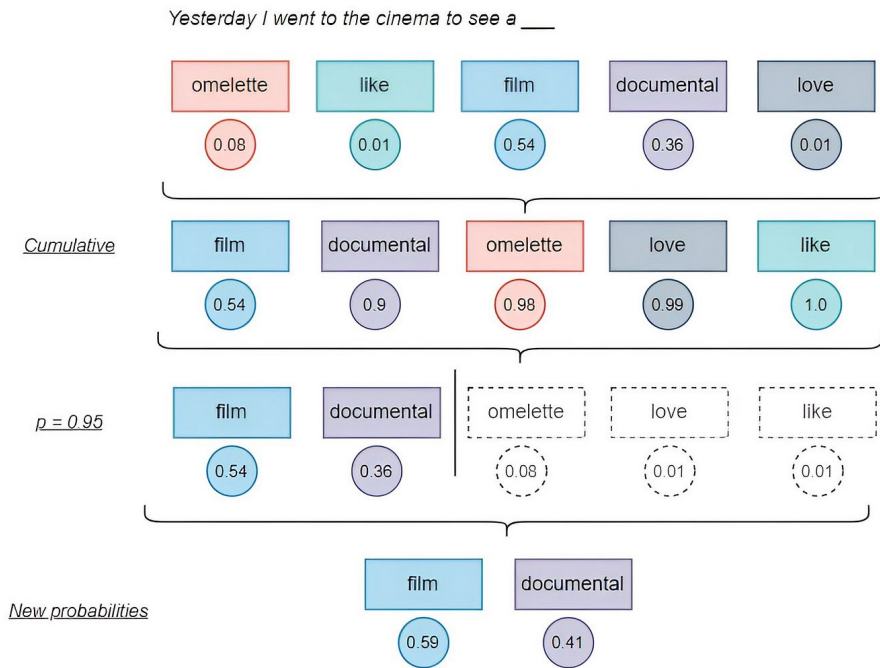
- Start with the default model, and switch to specialized models if needed
- Consider performance, quality, and compute trade-offs when choosing model size



Key Concepts: Top-p sampling

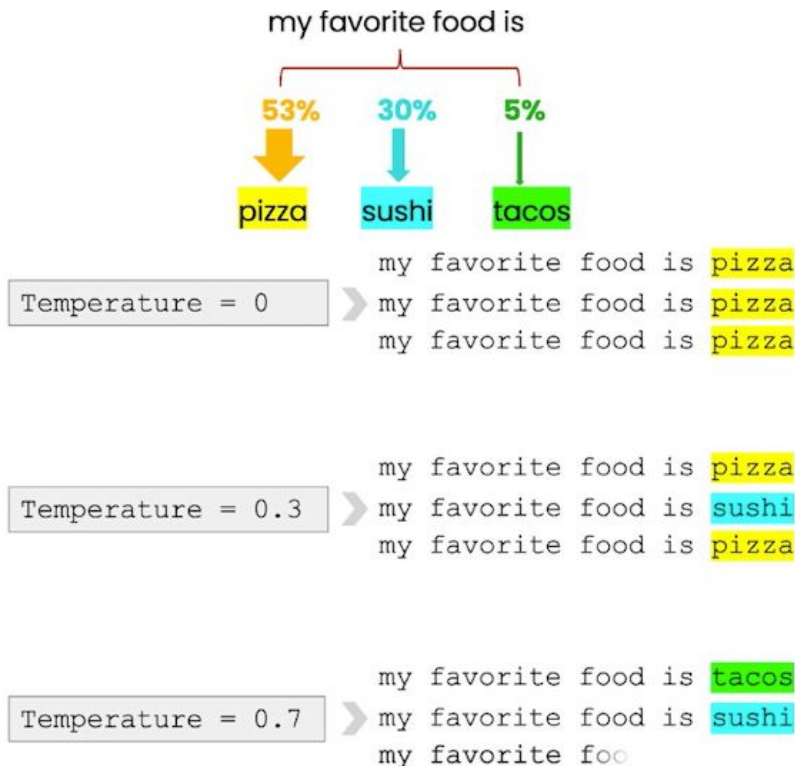
Top-k/Top-p sampling: Sampling methods to improve coherence

Top-k and top-p sampling are techniques that language models use to filter out low-probability word choices, improving the coherence and quality of the generated text by focusing on the most relevant and likely options.




Key Concepts: Temperature


Temperature is a parameter that controls the randomness or creativity in the text generated by an LLM. A higher temperature (e.g., 1.0) makes the output more random and diverse, while a lower temperature (e.g., 0.2) makes it more focused and deterministic.



Higher Temperature <> Higher Hallucination

Model: Claude 3 Sonnet | Temp: 1 | ⚙️ ☰

 explain about these the models on GrabGPT:
GPT-4, Vicuna, Claude 3 opus, Claude 3 sonnet, Claude 3 haiku, Dalle 3, Vision

 Sure, I'll explain the various models available on GrabGPT:

Claude 3 opus Claude 3 opus is an AI model from Anthropic, specifically designed for generating poetry in the opus (long poem) form. It can create intricate and structured poetic compositions.

Claude 3 sonnet Claude 3 sonnet is another poetry model from Anthropic, but this one is focused on generating sonnets, a specific type of 14-line poem with a particular rhyme scheme and structure.

Claude 3 haiku Claude 3 haiku is an AI model trained to generate haikus, which are short Japanese poems consisting of three lines with a 5-7-5 syllable structure. This model is adept at capturing the essence of haikus with their brevity and imagery.



ChatGPT

Tech-overhyped
people

ChatGPT

may generate may generate

inaccurate information

inaccurate information

ChatGPT may generate
inaccurate information

ChatGPT always
generates accurate information

API

API

1. **What is an API**
2. **API Programming**
3. **Best practices with GPT**

What is an API

- API (Application Programming Interface): A set of rules, protocols, and tools for building software and applications. An API specifies how software components should interact and allows different software applications to communicate with each other.
- **How APIs Work**
 - **Analogy:** Think of an API as a waiter in a restaurant. The kitchen (the system) prepares the food, and the waiter (API) delivers your order (request) to the kitchen and brings back the food (response) to your table (the client).
 - **Process:** A client sends a request to the API with specific parameters. The API processes the request, sends it to the server, retrieves the necessary data, and returns a response back to the client.
- **Why Use APIs?**
 - **Efficiency:** Automate tasks and streamline processes.
 - **Integration:** Enable interaction between different software systems.
 - **Innovation:** Facilitate the creation of new functionalities and services.

API Programming

- **Understand the API Documentation:** Review the documentation for the API you want to use to understand its functionalities, endpoints, and required parameters.
- **Authentication:** Many APIs require authentication (like API keys) to use their services. Ensure you have the necessary credentials.
- **Make a Request:** Use tools like Postman or write code in languages like Python, JavaScript, etc., to make requests to the API. This involves specifying the endpoint URL, setting request headers, and passing any required parameters. The 2 most common requests are **POST** and **GET**.
 - **POST:** To send data to a server to create/update a resource. The data is included in the body of the request. This may result in the creation of a new resource or the updates of existing resources or both.
 - **GET:** To retrieve information from the given server using a given URI (Uniform Resource Identifier). Requests using GET should only retrieve data and should have no other effect on the data.
- **Handle the Response:** Process the response returned by the API. This could involve parsing JSON or XML data and integrating it into your application.
- **Best libraries for Python API Programming:**
 - Fast API
 - Flask API

Using GPT API Locally

1. Request OpenAI API key from OpenAI website.
2. Create a **secrets.env** in your IDE.
3. Open text editor of your choice
4. Copy and paste the following:

```
OPENAI_API_KEY = #####
```

1. Save as **secrets.env** in your working folder
2. Load **secrets.env** with **python-dotenv**

Using Langsmith API Locally

1. Request the LangSmith API key from the [LangSmith website](#).
2. Store **secrets.env** in your IDE.
3. Open text editor of your choice
4. Copy and paste the following:

```
LANGCHAIN_TRACING_V2=true  
LANGCHAIN_ENDPOINT="https://api.smith.langchain.com"  
LANGCHAIN_API_KEY=  
LANGCHAIN_PROJECT=[Project name]
```

1. Save as **secrets.env** in your working folder
2. Load **secrets.env** with **python-dotenv**

Best Practices with GPT API

- Try to use something secure for your secrets!
- Do not upload your secrets.env into your GitHub repositories
- Do not share your API Key anywhere public!!
- Use a Virtual Environment!!!

Hands-On Setup

Prompt Engineering

Prompt Engineering

Zero-shot Prompting

- Providing just a task description or prompt to the LLM, **No examples or additional context** and relies on the LLM's general knowledge and understanding

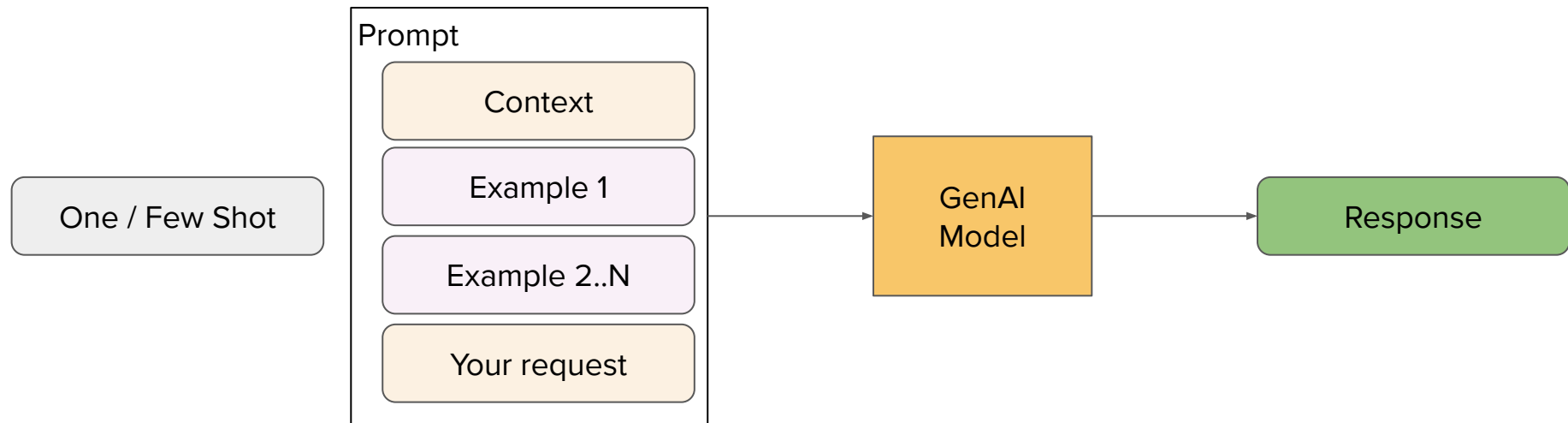
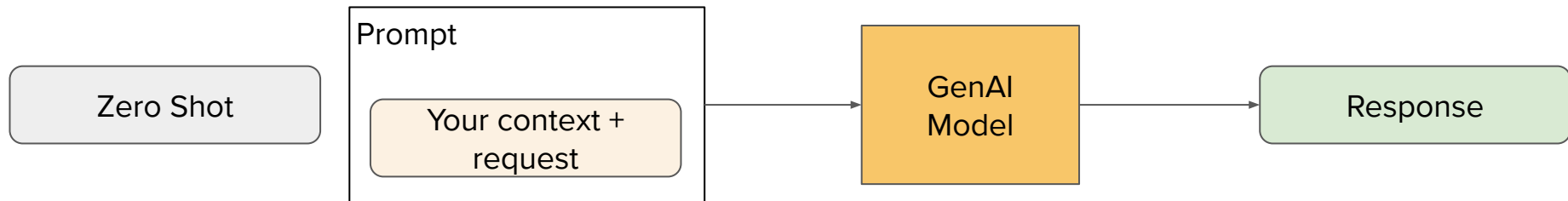
One-shot Prompting

- Providing **a single example** of the desired output along with the prompt to demonstrate the expected format and style which can improve the LLM's performance on specific tasks

Few-shot Prompting

- Providing **multiple examples** of the desired output along with the prompt which allows the LLM to better understand the task and variations. Generally leads to higher performance than zero-shot or one-shot

Prompt Engineering - Zero/One/Few shot



Prompt Engineering - Chain of Thought

One Shot Example

Input

Q:Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q:The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Output

A:The answer is 27



Chain of thoughts example

Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5+6=11$. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Output

A:The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23-20=3$. They bought 6 more apples, so they have $3+6=9$.
The answer is 9



GPT-4 Vision (Image to Text)

- GPT-4 Vision is an advanced AI model that can understand and generate both text and images. It's like having a super-smart assistant that can see and describe the world around it
- **Capabilities**
 - **Image Recognition:** It can identify objects, people, and scenes in images, similar to how you recognize things in photos.
 - **Image Description:** It can generate detailed descriptions of what's happening in an image, providing context and insights.
 - **Text and Image Integration:** It can combine information from text and images to give more comprehensive answers.

DALL-E 3 (Text to Image)

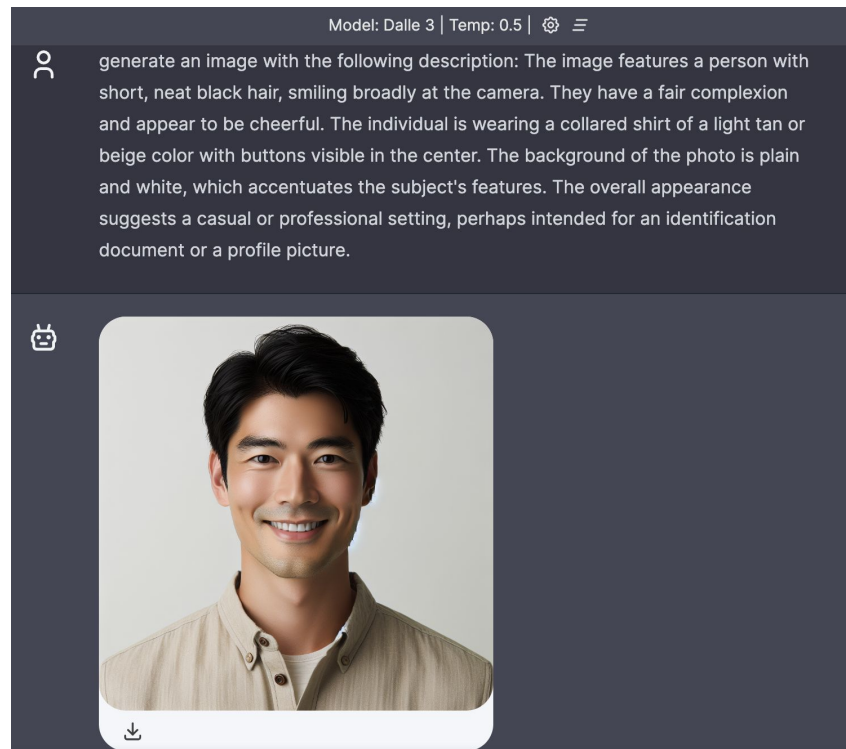
- DALL-E 3 is an AI model that generates high-quality images from textual descriptions. It can create vivid and detailed images based solely on the input text.
- **Capabilities**
 - **High-Resolution Images:** Produces images with high resolution and fine details, making them suitable for various professional and creative uses.
 - **Diverse Styles:** Can generate images in a wide range of artistic styles, from realistic to abstract, and mimic different artistic techniques.
 - **Complex Compositions:** Capable of creating intricate scenes and compositions, interpreting detailed and complex text prompts accurately.

Whisper (Speech to Text)

- Whisper is OpenAI's automatic speech recognition (ASR) system designed to transcribe and translate spoken language into text. It is capable of handling a variety of languages and dialects, providing accurate and reliable speech-to-text conversion.
- **Capabilities**
 - **Multilingual Support:** Can understand and transcribe speech in multiple languages, making it useful for global applications.
 - **Noise Robustness:** Designed to perform well even in noisy environments, ensuring accurate transcriptions in real-world conditions.
 - **High Accuracy:** Utilizes advanced neural network architectures to achieve high levels of transcription accuracy.

Create your own GrAvatar

- **What to do:**
 - Create your own Avatar by choosing the **Dalle3** model
 - Write a prompt with a description of yourself (or use gpt4-vision to write a description of yourself)
 - Generate and download the image
 - Post it on **#DSA** telegram channel
 - This will take a while so....

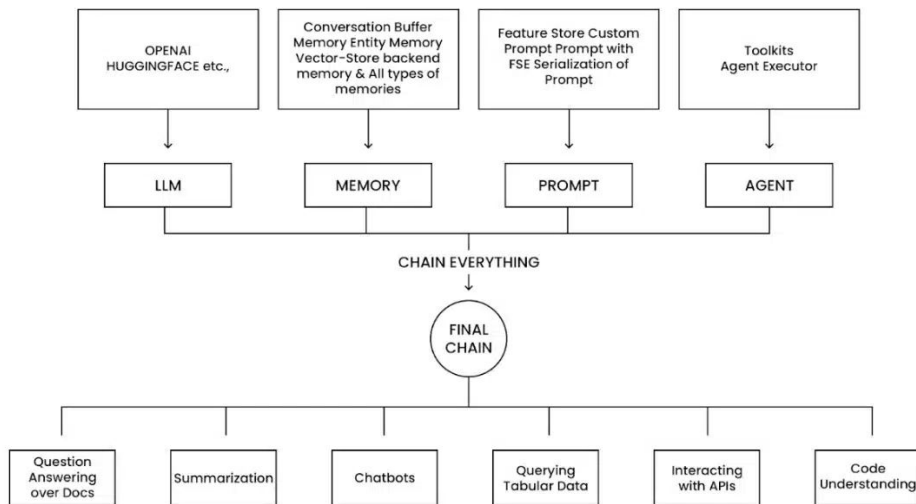


LangChain

What is LangChain?

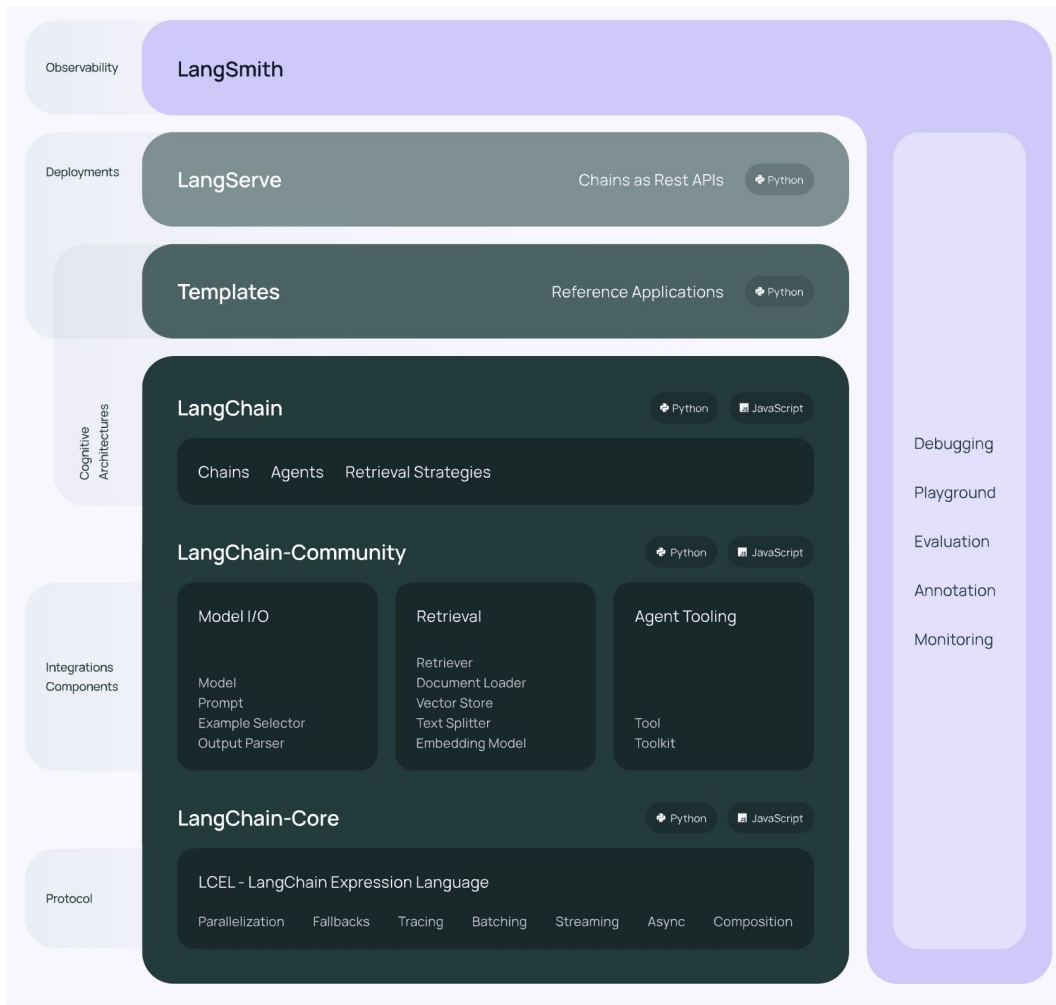
- Let's watch this video explaining what [langchain](#) is!
- LangChain is a framework that helps to build LLM apps written in Python, and JavaScript
- It leverages a complete set of modular building blocks to make it more production ready
- It utilizes a variety of LLMs, prompt templates, and chains that makes it suitable for your own use case
- It combines with LangSmith, for easier debugging and testing
- It combines with LangServe, for more observability

HIGH LEVEL STRUCTURE OF LANGCHAIN



LangChain Ecosystem

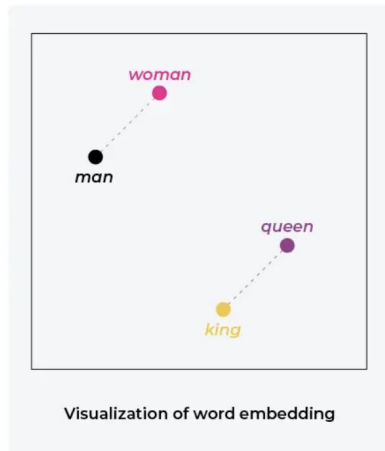
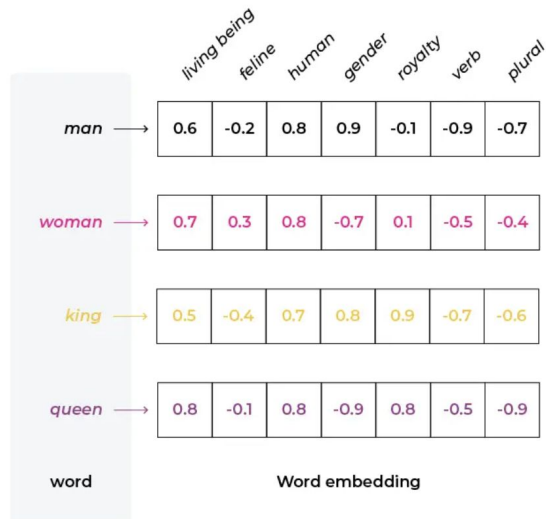
- We are able to utilize LangChain, LangSmith, and LangServe for a complete ecosystem
- This makes our LLM apps more production ready as we have complete traceability and complete monitoring of these apps



RAGs

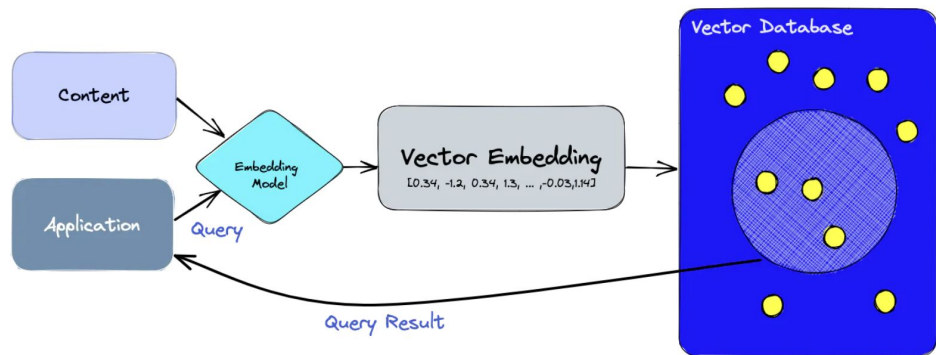
Embeddings

- Embeddings are words converted into a series of numbers (called **vectors**). These numbers capture the meaning and context of the words in a way that computers can understand.
- Words with similar meanings are located closer to each other
- Insights:
 - Similarity and Relationships
 - Analogies
 - Dimensionality



Vector Database

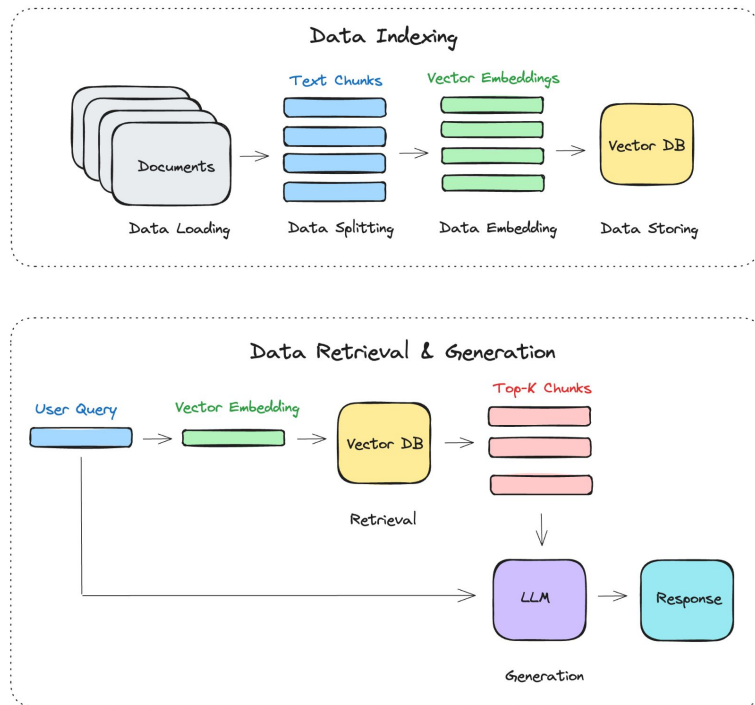
- A vector database is a specialized database designed to store, index, and query high-dimensional vector data efficiently.
- **Efficient Storage and Retrieval:** Vector databases store embeddings of text or other data and enable efficient retrieval of relevant information based on similarity search.
- **Embedding Indexing:** Embeddings generated from the input query are matched against pre-stored embeddings to find the most relevant documents or pieces of information.
- **Examples:**
 - FAISS
 - Milvus
 - [PGVector](#)



Retrieval Augmented Generation (RAG)

- Retrieval-Augmented Generation (RAG) is an advanced AI approach that combines the strengths of two techniques: **information retrieval** and **generative modeling**.
- **Information Retrieval:** This involves searching and fetching relevant documents or pieces of information from a large knowledge base. It's like finding the most relevant pages in a library when you have a specific query.
- **Generative Modeling:** This involves creating new text based on input data, similar to how chatbots generate responses. Generative models can provide detailed, coherent, and contextually appropriate answers by understanding and producing human-like text.

Basic RAG Pipeline





We are in the endgame now.

Uni/Multimodal

LLMs

(GPT4-o, Vision, DALLE3,
Whisper)

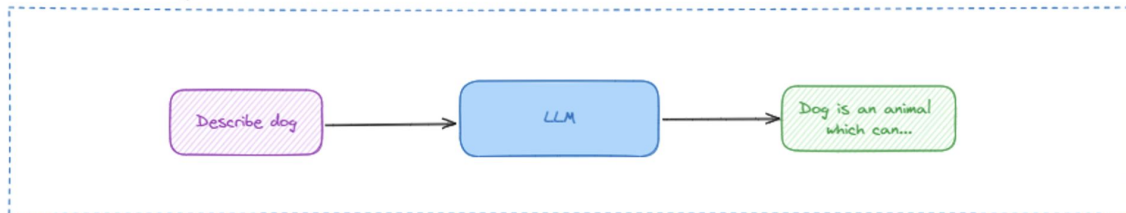


OpenAI Flagship Model GPT4-o

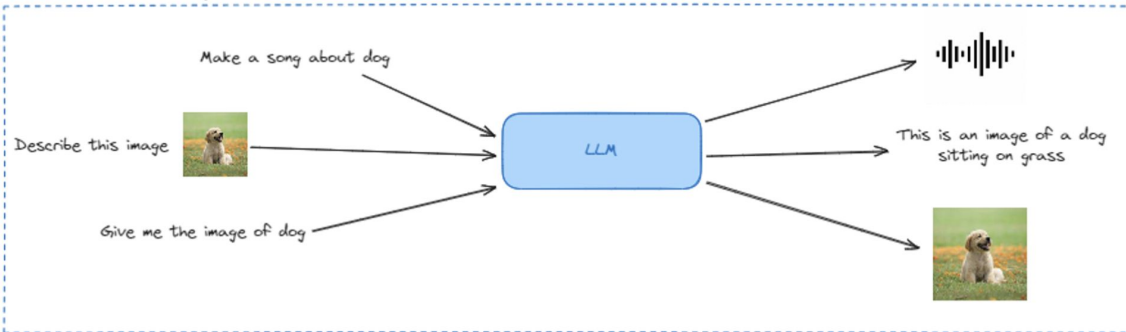
Multimodal LLMs

- Multimodal Large Language Models (MLLMs) are advanced AI systems designed to understand and generate human-like responses across various types of data inputs, including text, images, audio, and video.
- Unlike traditional LLMs, which are primarily text-based, multimodal LLMs integrate and process information from multiple **modalities** to perform more complex tasks and provide richer, more contextually aware outputs.

Unimodal Learning



Multimodal Learning



ThankYou!!

I saw a great movie about databases the other day...What am I excited about?

- Can't wait for the SQL