

NOVA

IMS

Information
Management
School

7

LARGE LANGUAGE MODELS

Business Cases with Data Science

© 2020-2025 Nuno António and Hugo Silva

Acreditações e Certificações



Summary

- 
1. Introduction
 2. Generative AI
 3. Prompt engineering
 4. Retrieval-augmented generation
 5. Parameters tuning
 6. ChatGPT: Getting an API Key
 7. Azure Open AI

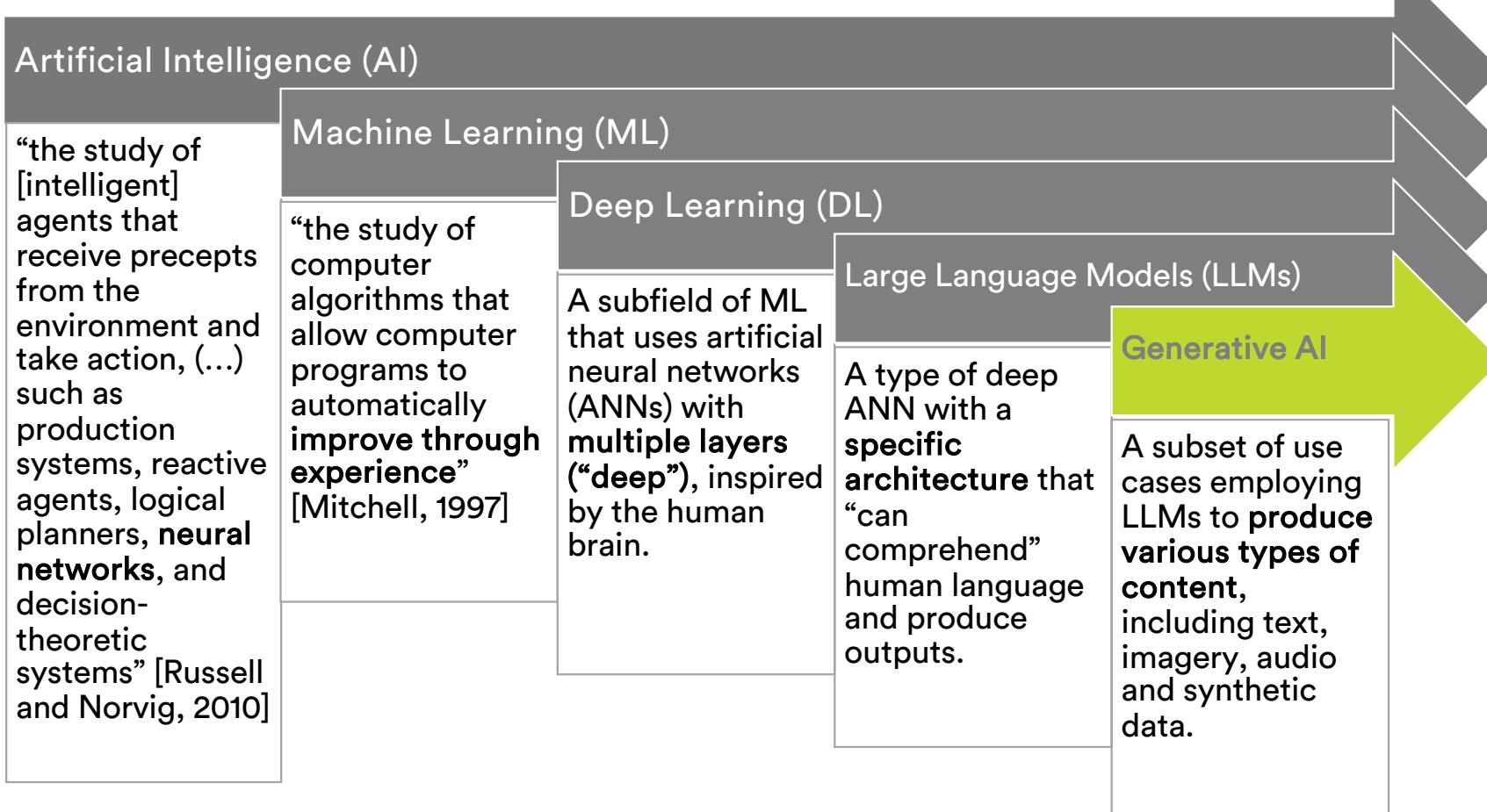
7.1

Introduction

From Neural Networks to Large Language Models

Evolution of Machine Learning to Generative AI

The foundations that led to the development of Large Language Models (LLMs).



Generative AI context

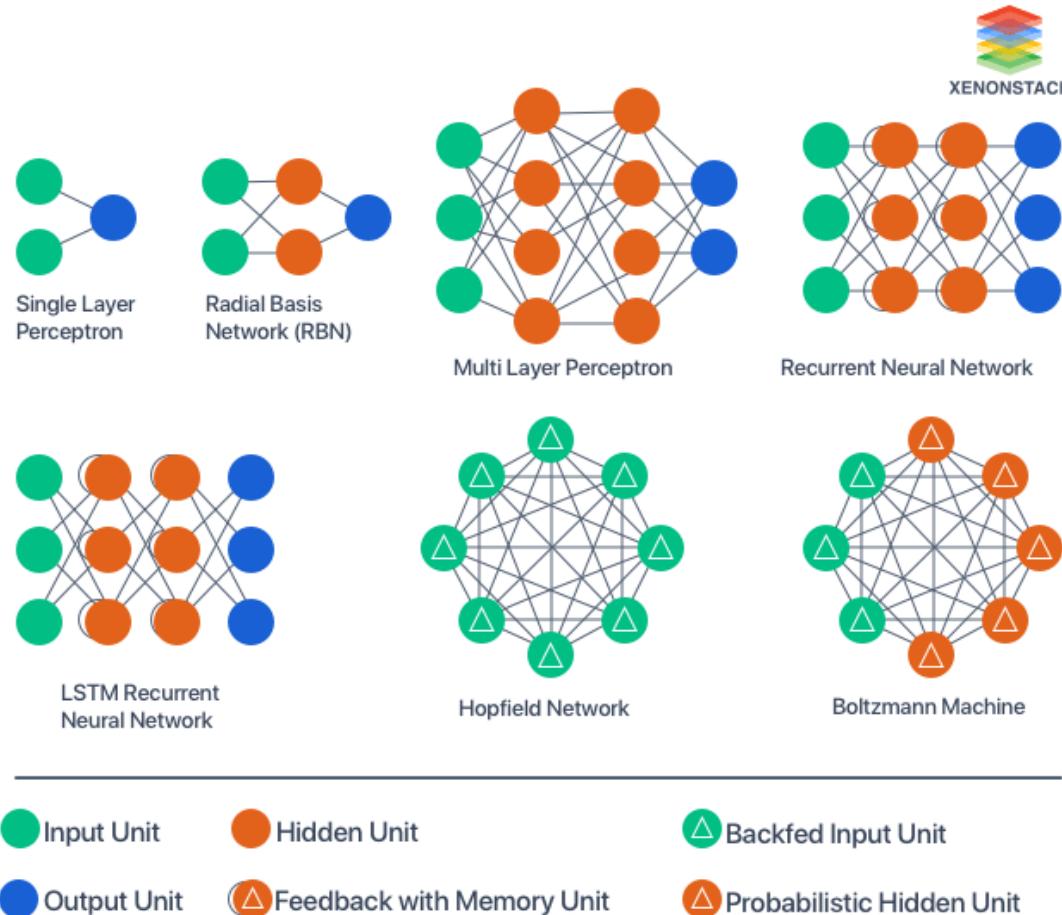
WHAT is Generative AI? A subset of artificial intelligence focused on the ability of machines to create outputs across various modalities (e.g., text, images, audio, code, voice, video)

HOW does it work? It uses LLMs, such as OpenAI's GPT, trained on massive amounts of data to understand human communication and natural language

WHY now? Innovations in hardware, cloud-native stack, software engineering, machine learning, and deep learning allowed the creation of Generative AI solutions

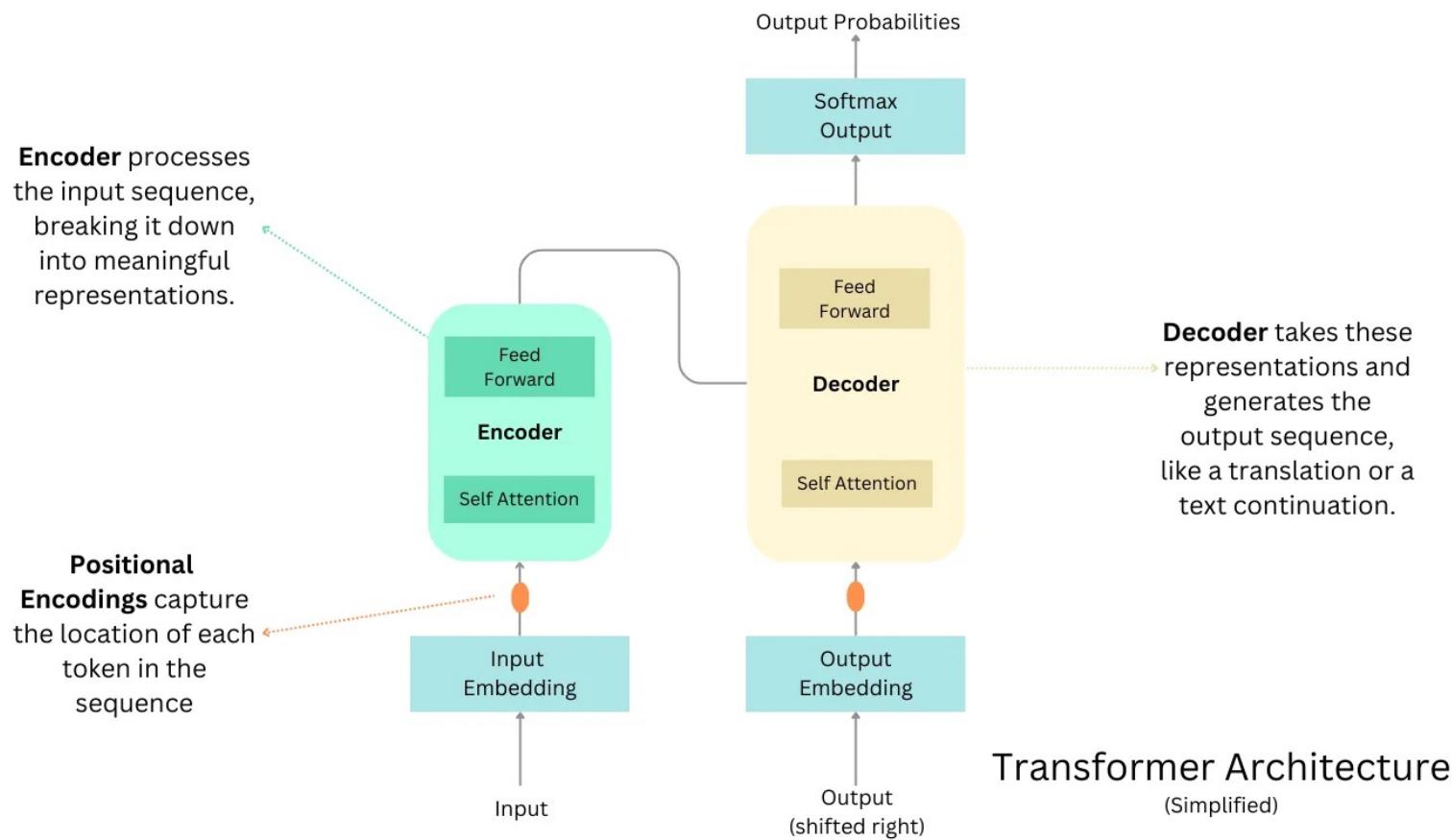
WHO is involved? Technology leaders and start-ups are developing user-facing applications on these underlying models, with apps such as ChatGPT, and Dall-E rising in popularity

The roadmap to LLMs: multiple architectures



source: <https://www.tomorrow.bio/post/nodes-that-know-how-neural-network-architecture-learns-2023-06-4669541450-ai>

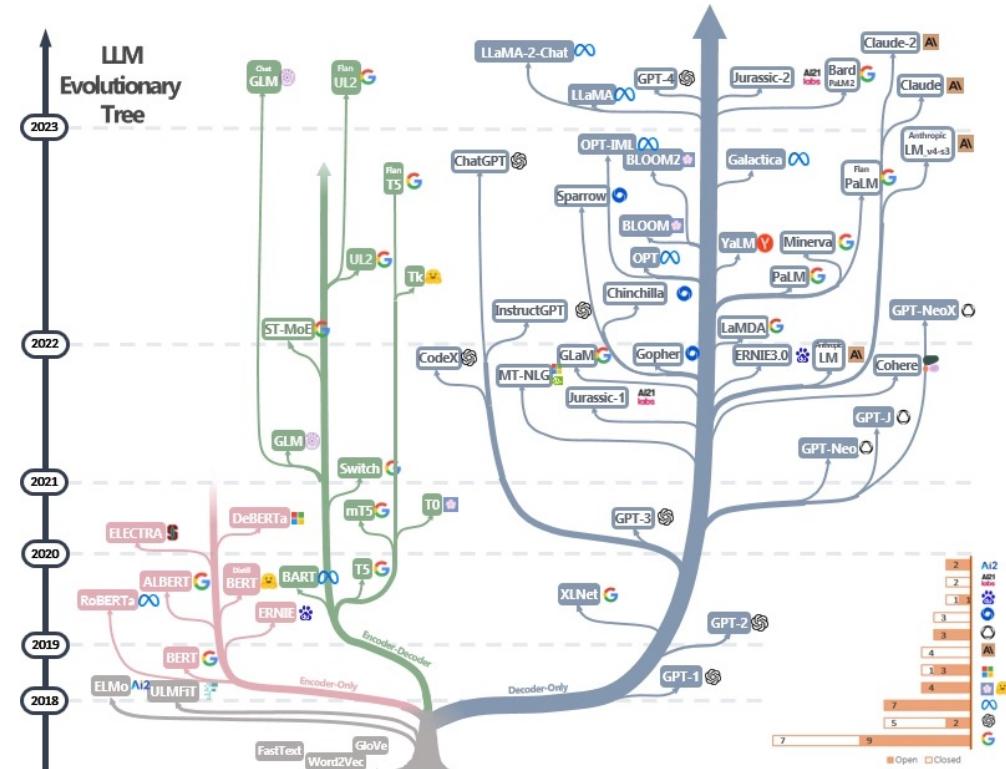
Transformer: the game changer



source: <https://medium.com/@tech-gumptions/transformer-architecture-simplified-3fb501d461c8>

LLM model types and evolution (1/3)

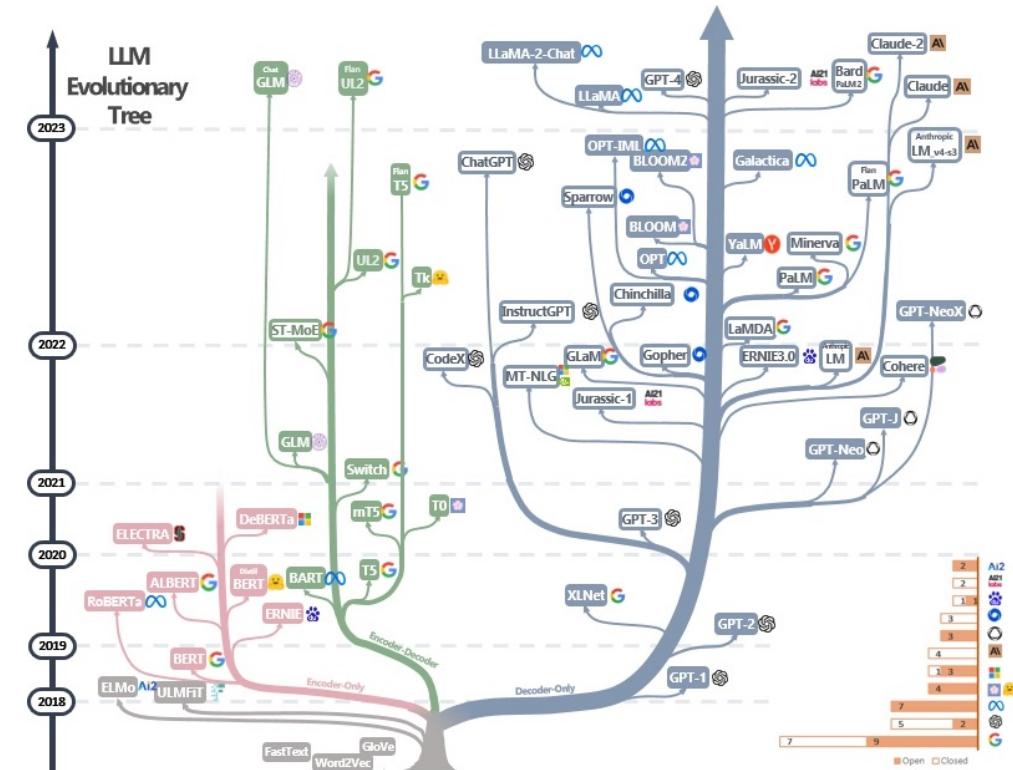
Encoder-decoder: Architecture with two parts. The encoder processes the input (e.g., English sentence) and compresses it in a context. The decoder takes the context and produces an output (e.g., a translation to French)



Source: <https://github.com/Mooler0410>

LLM model types and evolution (2/3)

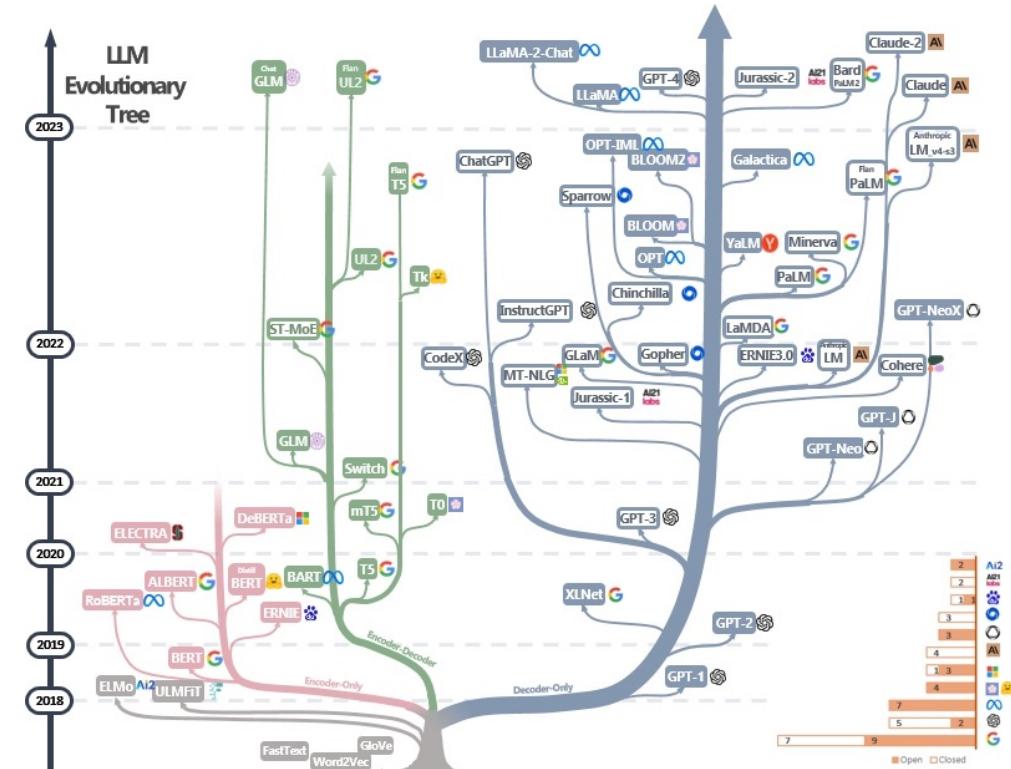
Encoder-only: Architecture with just the encoder. It processes the input and produces a direct output. Useful for tasks where you do not need a transformation into another “type” or “form” (e.g., classification tasks)



Source: <https://github.com/Mooler0410>

LLM model types and evolution (3/3)

Decoder-only: Architecture with only the encoder. It starts with basic information and expands or generates output based on it. This what models like GPT use (generate stories, answers, etc.)



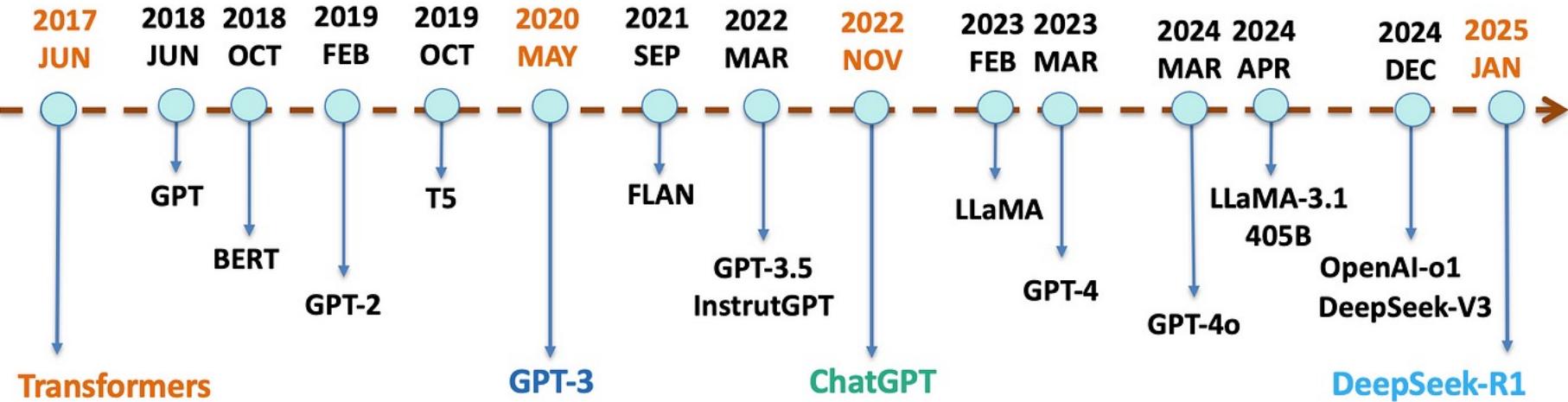
Source: <https://github.com/Mooler0410>

7.2

Generative AI

LLMs as enablers of the Generative AI

The evolution of LLMs



source: medium.com

LLMs are undergoing rapid evolution, making them applicable across diverse contexts and realities...



Thousands of open-source LLMs are available



Hundreds of lighter versions of models, that require less powerful hardware, facilitating broader accessibility and ease of implementation



Dozens of fine-tuned versions of the main models

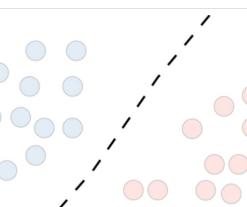
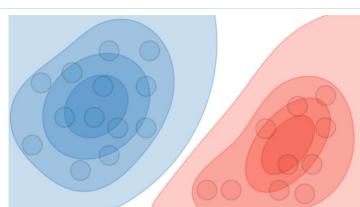
Types of Deep Learning Models

Discriminative

- Used to classify or predict
- Typically trained with labeled data
- Learns the relationship between the features of the data points and the labels

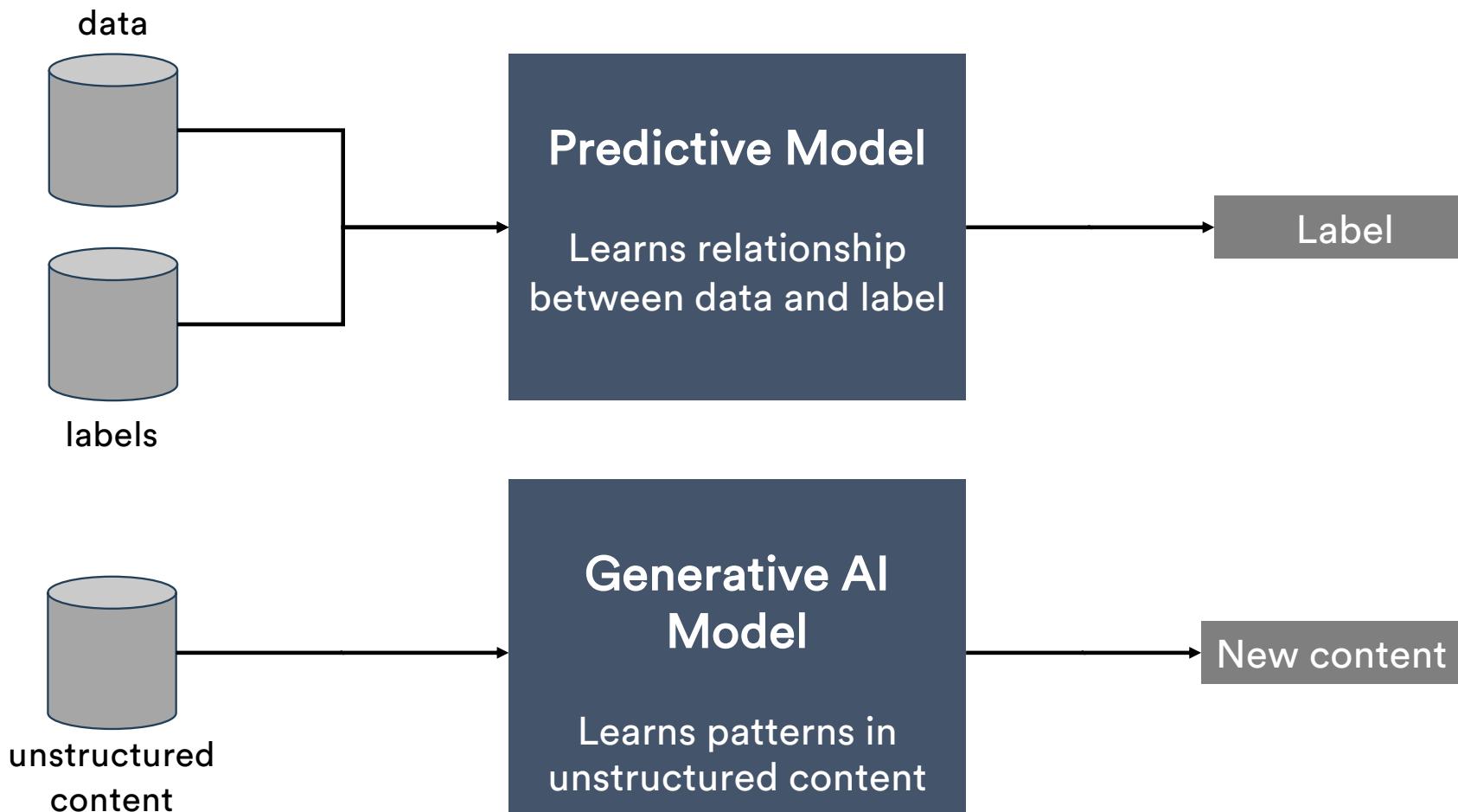
Generative

- Generates data points similar to the ones it was trained with
- Understand the data distribution and how likely a given example is
- Predict next “word” in a sentence

	Discriminative model	Generative model
Goal	Directly estimate $P(y x)$	Estimate $P(x y)$ to then deduce $P(y x)$
What's learned	Decision boundary	Probability distributions of the data
Illustration		
Examples	Regressions, SVMs	GDA, Naive Bayes

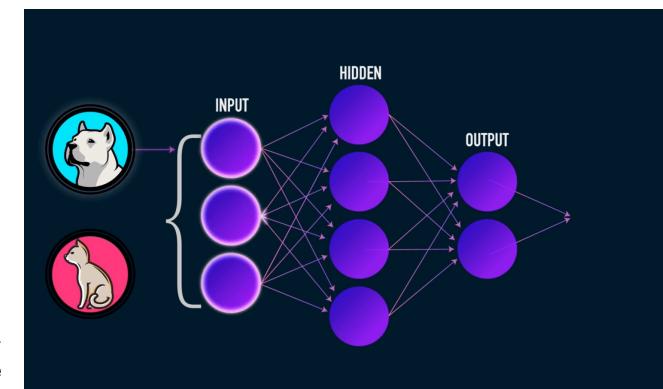
source: stanford.edu

Predictive vs Generative Models



LLMs sizes

- LLMs are measured by the number of **parameters** in the model and the number of **tokens** they were trained on
- **Parameters** are the weights and biases connecting nodes in a neural network
- **Tokens** are how LLMs break up sentences into “pieces”. These can be multiple words, single words, or parts of words
- Generally, the more parameters a model has and the more tokens it was trained, the better it performs



Source: <https://towardsdatascience.com/visualizing-artificial-neural-networks-annts-with-just-one-line-of-code-b4233607209e>

Text tokenization (1/3)

- Foundational step in Natural Language Processing (NLP) processing, including in LLMs like GPT-4 and Llama-2
- Involves breaking text into smaller chunks (tokens), which can be as short as one character or as long as one word (in some cases multiple words)
- Tokens are then processed and used as input for Machine Learning models

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

GPT-3.5 & GPT-4 GPT-3 (Legacy)

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 

Sequences of characters commonly found next to each other may be grouped together: 1234567890

[Clear](#) [Show example](#)

Tokens	Characters
57	252

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 000000

Sequences of characters commonly found next to each other may be grouped together: 1234567890

TEXT TOKEN IDS

A helpful rule of thumb is that one token generally corresponds to ~4 characters of text for common English text. This translates to roughly $\frac{3}{4}$ of a word (so 100 tokens \approx 75 words).

Text tokenization (2/3)

GPT-3.5 & GPT-4 GPT-3 (Legacy)

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 

Sequences of characters commonly found next to each other may be grouped together: 1234567890

[Clear](#) [Show example](#)

Tokens	Characters
57	252

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 000000

Sequences of characters commonly found next to each other may be grouped together: 1234567890

TEXT TOKEN IDS

A helpful rule of thumb is that one token generally corresponds to ~4 characters of text for common English text. This translates to roughly $\frac{3}{4}$ of a word (so 100 tokens \approx 75 words).

GPT-3.5 & GPT-4 GPT-3 (Legacy)

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 

Sequences of characters commonly found next to each other may be grouped together: 1234567890

[Clear](#) [Show example](#)

Tokens	Characters
57	252

[8607, 4339, 2472, 311, 832, 4037, 11, 719, 1063, 1541, 956, 25, 3687, 23936, 382, 35020, 5885, 1093, 100166, 1253, 387, 6859, 1139, 1690, 11460, 8649, 279, 16940, 5943, 25, 11410, 97, 248, 9468, 237, 122, 271, 1542, 45045, 315, 5885, 17037, 1766, 1828, 311, 1855, 1023, 1253, 387, 41141, 3871, 25, 220, 4513, 10961, 16474, 15]

TEXT TOKEN IDS

A helpful rule of thumb is that one token generally corresponds to ~4 characters of text for common English text. This translates to roughly $\frac{3}{4}$ of a word (so 100 tokens \approx 75 words).

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

Text tokenization (3/3)

You can use the tool below to understand how a piece of text might be tokenized by a language model, and the total count of tokens in that piece of text.

GPT-4o & GPT-4o mini GPT-3.5 & GPT-4 GPT-3 (Legacy)

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🖐

Sequences of characters commonly found next to each other may be grouped together: 1234567890

[Clear](#) [Show example](#)

Tokens	Characters
54	252

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🖐

Sequences of characters commonly found next to each other may be grouped together: 1234567890

[Text](#) [Token IDs](#)

Note: Your input contained one or more unicode characters that map to multiple tokens. The output visualization may display the bytes in each token in a non-standard way.

A helpful rule of thumb is that one token generally corresponds to ~4 characters of text for common English text. This translates to roughly ¼ of a word (so 100 tokens ≈ 75 words).

You can use the tool below to understand how a piece of text might be tokenized by a language model, and the total count of tokens in that piece of text.

GPT-4o & GPT-4o mini GPT-3.5 & GPT-4 GPT-3 (Legacy)

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🖐

Sequences of characters commonly found next to each other may be grouped together: 1234567890

[Clear](#) [Show example](#)

Tokens	Characters
54	252

```
[12488, 6391, 4014, 316, 1001, 6602, 11, 889, 1236, 1700, 1573, 25, 3862, 181386, 364, 61064, 9862, 1299, 166700, 1340, 413, 12648, 1511, 1991, 20290, 15683, 290, 27899, 11643, 25, 93643, 248, 52622, 122, 279, 168191, 328, 9862, 22378, 2491, 2613, 316, 2454, 1273, 1340, 413, 73263, 4717, 25, 220, 7633, 19354, 29338, 15]
```

[Text](#) [Token IDs](#)

A helpful rule of thumb is that one token generally corresponds to ~4 characters of text for common English text. This translates to roughly ¼ of a word (so 100 tokens ≈ 75 words).

If you need a programmatic interface for tokenizing text, check out our [tiktoken](#) package for Python. For JavaScript, the community-supported [@dbdq/tiktoken](#) package works with most GPT models.

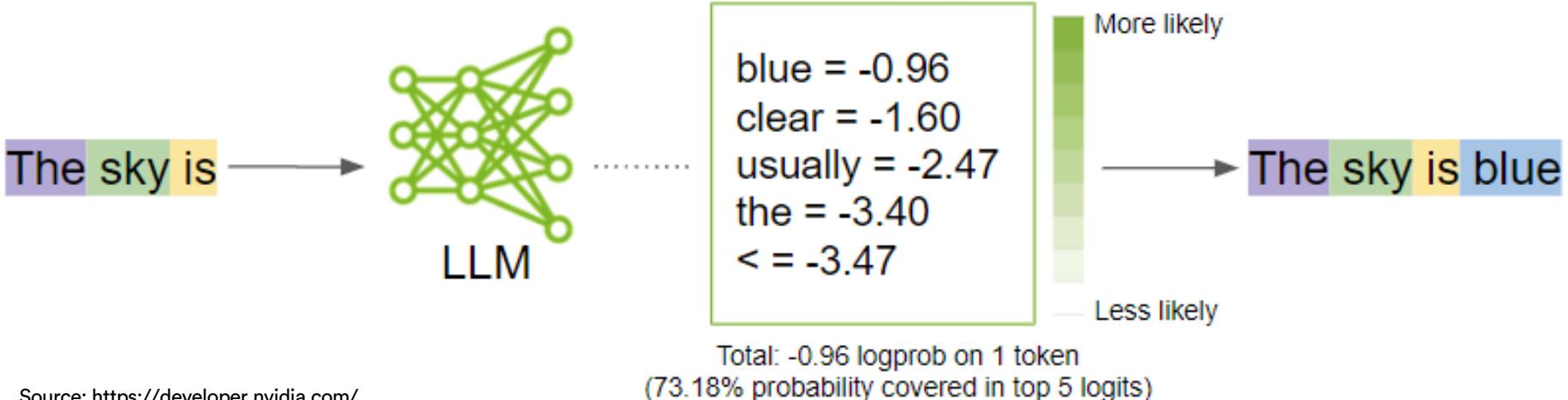
Source: <https://platform.openai.com/tokenizer>

LLMs are autoregressive language models

LLMs take an input text and repeatedly predict the next token based on the context.

$$P(token_k | token_{context}) = \frac{\exp(logit_k)}{\sum_j \exp(logit_j)}$$

This is the probability of $token_k$ given the context from previous tokens ($token_1$ to $token_{k-1}$) and $logit_k$ is the output of the neural network.



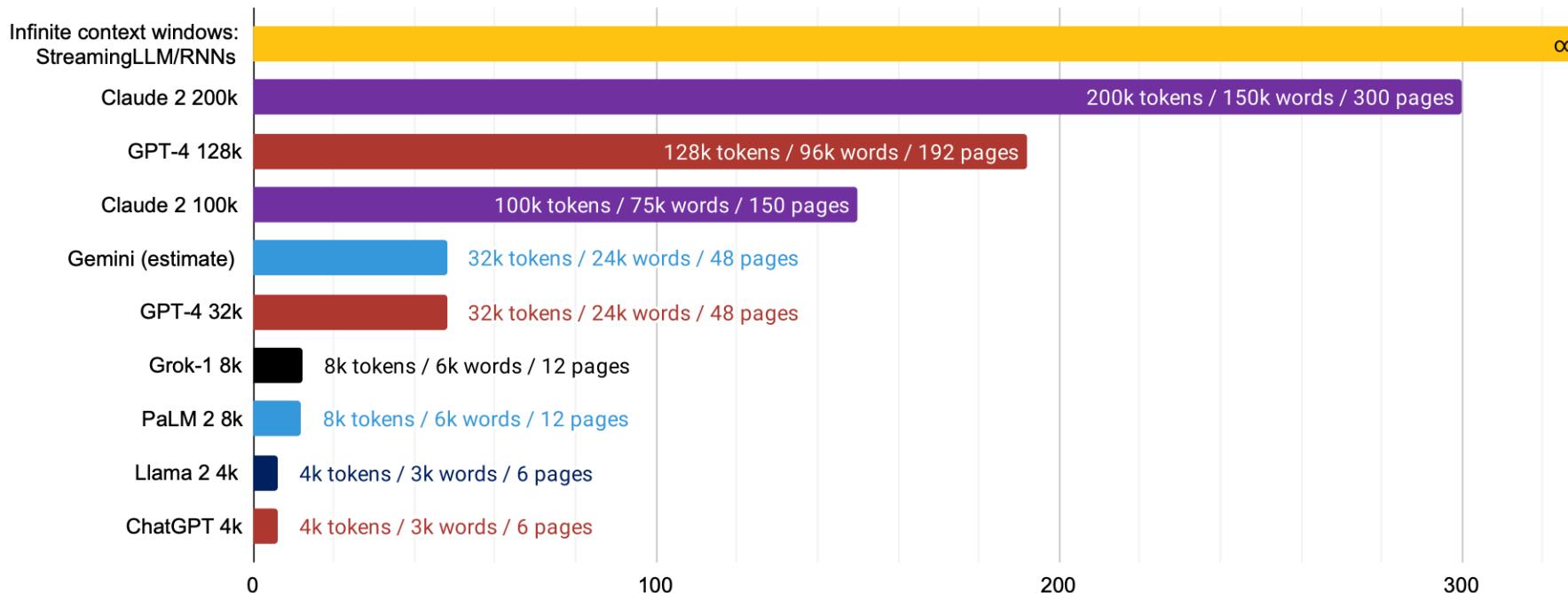
Source: <https://developer.nvidia.com/>

LLMs context window

- The context window (or “context length”) of an LLM is the amount of text, in tokens, that the model can consider or “remember” at any one time. A larger context window enables an AI model to **process longer inputs and incorporate more information into each output.**
- Increasing an LLM’s context window size translates to increased accuracy, fewer hallucinations, more coherent model responses, longer conversations, and an improved ability to analyze longer sequences of data.
- However, increasing context length is not without tradeoffs: it often entails increased computational power requirements, which in turn increases costs.

LLMs context size

2023 CONTEXT WINDOWS (MAX IN/OUT LENGTH)

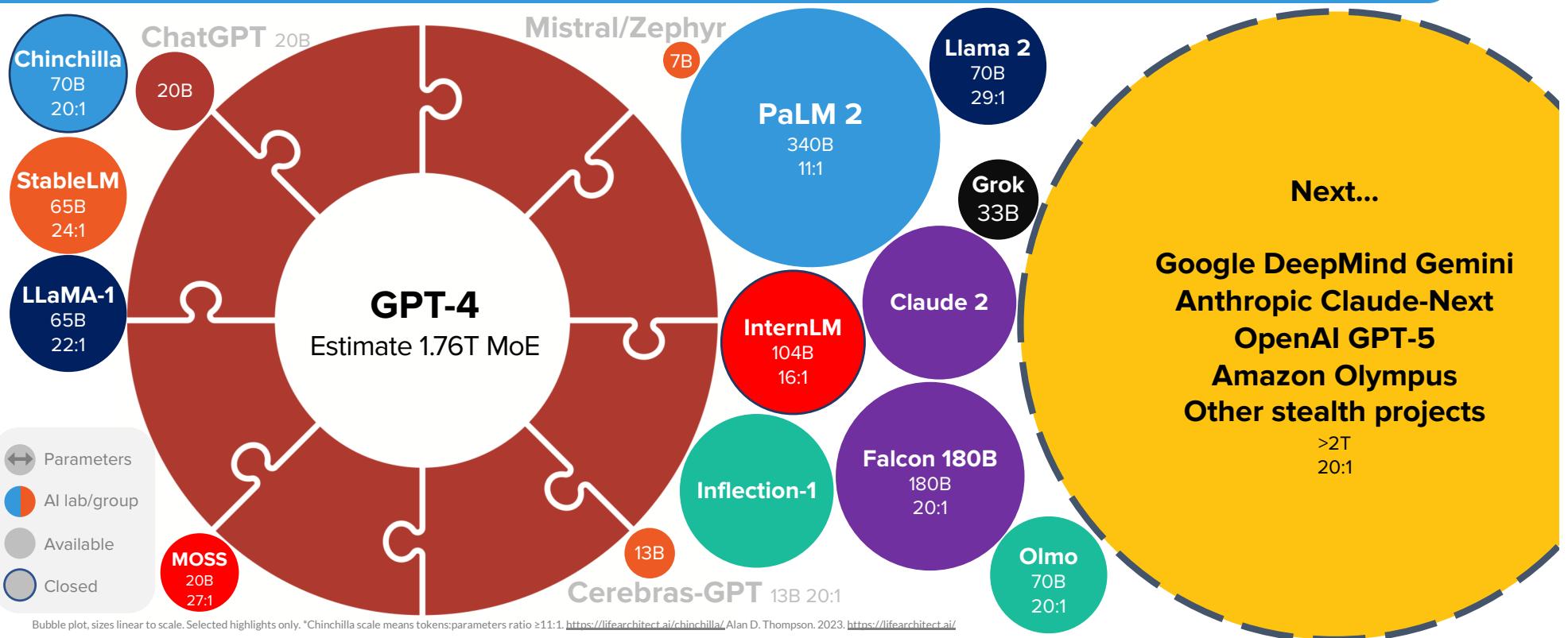


Using rounded figures of 1 token = 0.75 words (e.g. 32,000 tokens ≈ 24,000 words), 500 words ≈ 1 page. Alan D. Thompson, November 2023. <https://lifearchitect.ai/models/>



LifeArchitect.ai/models

2023-2024 OPTIMAL LANGUAGE MODELS

NOV/
2023
LifeArchitect.ai/models

Generative LLMs

AI Titans Face Off

	GPT-4o	Gemini 1.5 001	Llama 3.1 7B	Mistral 7B	Claude 3.5	DeepSeek V3
Model Overview						
Country	USA	USA	USA	France	USA	China
Developer	OpenAI	Google	Meta	Mistral AI	Anthropic	DeepSeek Inc.
Release Date (model released)	2023	2024	2024	2023	2023	2023
Capabilities and Features						
Supported Modalities	Text, Image	Text, Image, Video, Voice	Text	Text	Text, Image	Text
Capabilities	Excels in coding, complex tasks, versatile applications	Multimodal with advanced reasoning, data processing	Strong in language math with improved generation	Specialized in language tasks with open-source efficiency	Safe, nuanced text generation ethical AI alignment	Specialized in scientific research and data analysis
Training and Updates						
Training Data	Extensive web data	Proprietary data	Large web data	Open web resources	Open web resources	Proprietary datasets
Ethical Considerations	Bias, Misinformation	Bias, Privacy concerns	Bias, Privacy concerns	Bias, Privacy concerns	Strong focus on safety	Bias, Privacy concerns
Knowledge Updates	10/2023	11/2023	12/2023	Unknown	08/2023	Unknown
Performance Metrics (Tokens)						
Number of Tokens Supported	128K	1M	128K	32K	200K	128K
Tokens That Can Be Generated	16.4K	8K	2K	8K	4K	8K
Accessibility and Licensing						
Open Source	No	No	Yes	Yes	No	Yes
API Providers	OpenAI Azure OpenAI	Google AI Studio Vertex AI	Azure AI AWS Bedrock Google AI Studio Vertex AI IBM WatsonX DeepInfra	Azure AI AWS Bedrock Google AI Studio Vertex AI Snowflake Cortex	Anthropic, AWS Bedrock Google AI Studio Vertex AI	DeepSeek HuggingFace
Pricing and Licensing						
Cost/Licensing	Subscription, Pay-per-use	Subscription, Pay-per-use	Open-source, free to use	Open-source, free to use	Open-source, free to use	Details Unclear
Input Cost (per Million token)	\$2.50	\$0.13	\$0.23	\$0.25	\$15.00	\$0.14
Output Cost (per Million token)	\$10.00	\$0.38	\$0.40	\$0.25	\$75.00	\$0.28

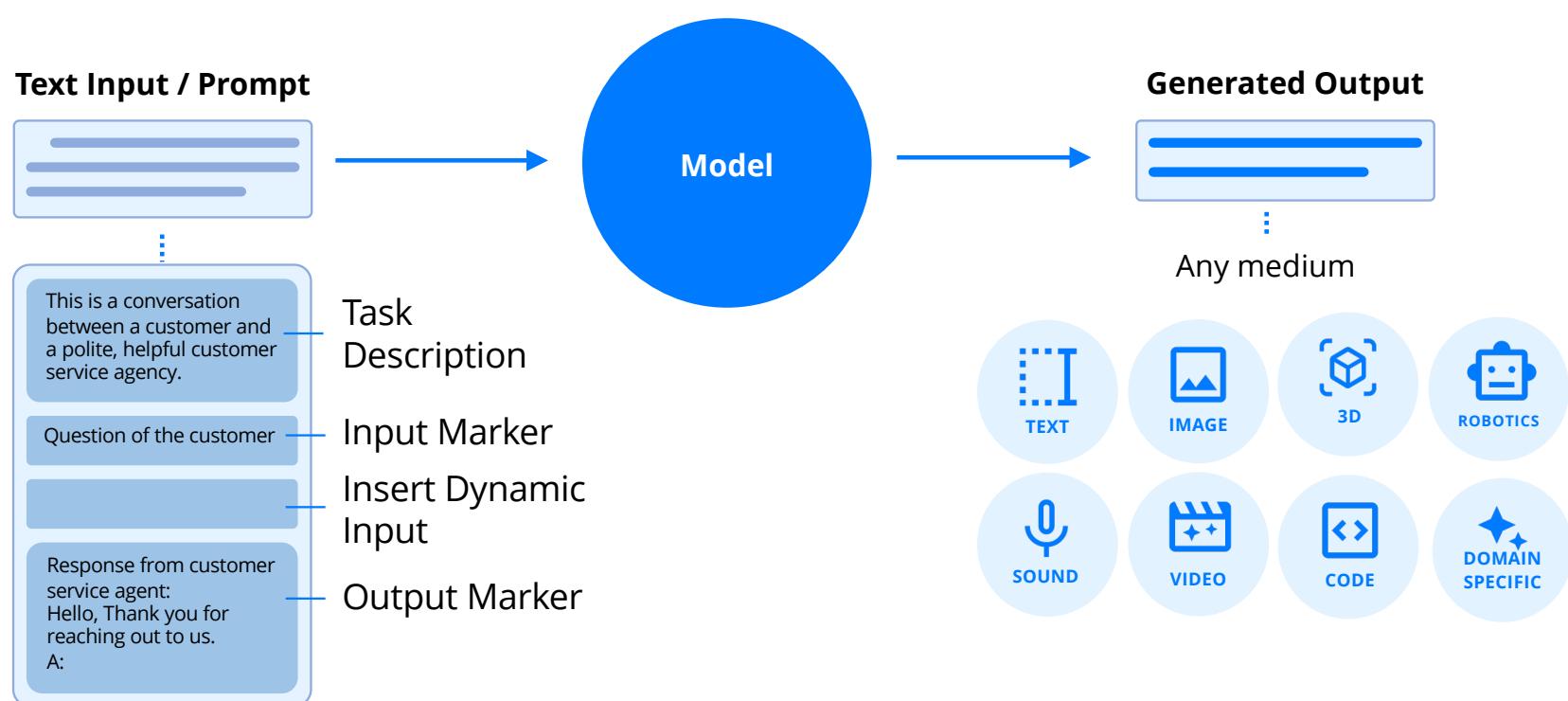


Ghazi MEJAT
Daily AI content

Use the right AI for the right task

REPOST

How Generative AI applications work?



How models adapt to new domains?

Depending on the use case, models may need to adapt to new domains or be exposed to new data. The implementation efforts (time, cost and compute effort) are completely different and may impact the decision.

These are typically the focus of most organization

IMPLEMENTATION EFFORT

Build your own LLM from scratch

Train a model from the ground with domain specific data

Very in-dept data domain specific and model trained on raw data.

Very High
Medium to High
Low to Medium



AUTO CLAIM AUTOMATIC DAMAGE ASSESSMENT



AUTOMATED CONTENT GENERATION
(e.g., market summaries; investment advice)

Finetune existing Foundational Model

Train an existing model with domain specific data

Use of an existing baseline model and tune-it for specific purposes with supervised training data

Very High
Medium to High
Low to Medium



ENHANCED CREDIT SCORING & UNDERWRITING



ENHANCED CHATBOT & CUSTOMER SUPPORT

In-Context Learning with RAG

Feed the LLMs with private contextual data as part of the prompts

Use of an existing baseline model and tune-it for specific purposes with supervised training data

Very High
Medium to High
Low to Medium



USE CUSTOMER DATA AS INPUT TO EXTRACT RELEVANT INFORMATION AND PROVIDE CONTEXTUAL RESPONSES.

7.3

Prompt engineering

Designing inputs to produce optimal outputs

Prompt engineering

- It is the process of transmitting instructions to LLMs
- A prompt is a Natural Language text describing the task the model should perform
- A prompt can be:
 - A query (e.g., “What is the Theory of relativity?”)
 - A command (e.g., “Write a post for NOVA IMS Instagram”)
 - A role assignment (e.g., Act as a Portuguese native speaker)
 - A statement on feedback (e.g., “answer in a formal way”)
 - A statement on context (e.g., providing a text to ask for a summarization)
- Creating prompts requires creativity and attention to detail
- Prompts should be as clear and specific as possible

Why being specific is important



https://www.youtube.com/watch?v=cDA3_5982h8&t=13s

Types of operations



Prompt Engineering Guide

There are 3 main types of prompt operations that can be performed:



REDUCTIVE PROMPT OPERATIONS

Takes a large amount of text and makes it shorter

- Summarizations (executive summary, translate text to bullets, etc.)
- Extractions (retrieve data from a text, example: extract dates, names, etc.)
- Characterizations (describe a specific text)
- Analyzing (find text patterns)
- Critic (provide contextual feedback on a text, as sentiment analysis.)



TRANSFORMATION PROMPT OPERATIONS

Take a given input and transform it

- Reformatting (change the text output format, example: JSON to YAML or to XML)
- Language translations (translate text from English to Portuguese)
- Code Languages Translations (example convert COBOL to Python)
- Restructuring (reorganize code or make it more performance efficient)
- Tone Modification (change the communication output to be more formal)



GENERATIVE OPERATIONS

Create new, unseen content

- Planning (draft a vacations plan given a list of locations)
- Drafting (write an introduction for a letter or a draft plot for a fiction book)
- Brainstorm (imagine a list of possibilities given a particular use case)
- Expansion (generate more content given a short text input)

Basics of prompt engineering (1/3)

Be specific and clear:

- Use delimiters (“, -, #, <, >) to split the different sections of your prompt (e.g., defining steps or framing a user message)
- Define the output format. If JSON or XML, give an example if possible
- Be clear in assumptions and conditions. For example, the return value should be from a list of three possible values
- Give a few input and output examples. This is called few-shot prompting

Basics of prompt engineering (2/3)

Push the model to think about the answer:

- Give step-by-step instructions to force it to go through all the steps
- If the previous approach is not viable, break the task into smaller sequential prompts

Basics of prompt engineering (3/3)

“LMs are 90% correct and 100% confident”

Be aware of hallucinations:

- Ask the model to link the answer to relevant information from the context, then answer the question based on the data
- If the model uses the Internet, ask for URL references
- If running from the API, reduce “temperature” to a low value

Zero-shot prompting

Q: Write an answer to the following customer complaint with at most 300 characters.

###

Complaint: This is one of the worst companies I've ever done business with. They don't answer the phones and don't care about the customer.

A: We're sorry to hear about your experience. We strive to provide excellent customer service and regret any inconvenience. Please email us at support@example.com with your contact details, and we'll ensure prompt assistance.

Few-shot prompting

Q: Considering the examples below, write an answer to the following customer complaint with at most 300 characters:

###

Complaint: This is one of the worst companies I've ever done business with. They don't answer the phones and don't care about the customer.

###

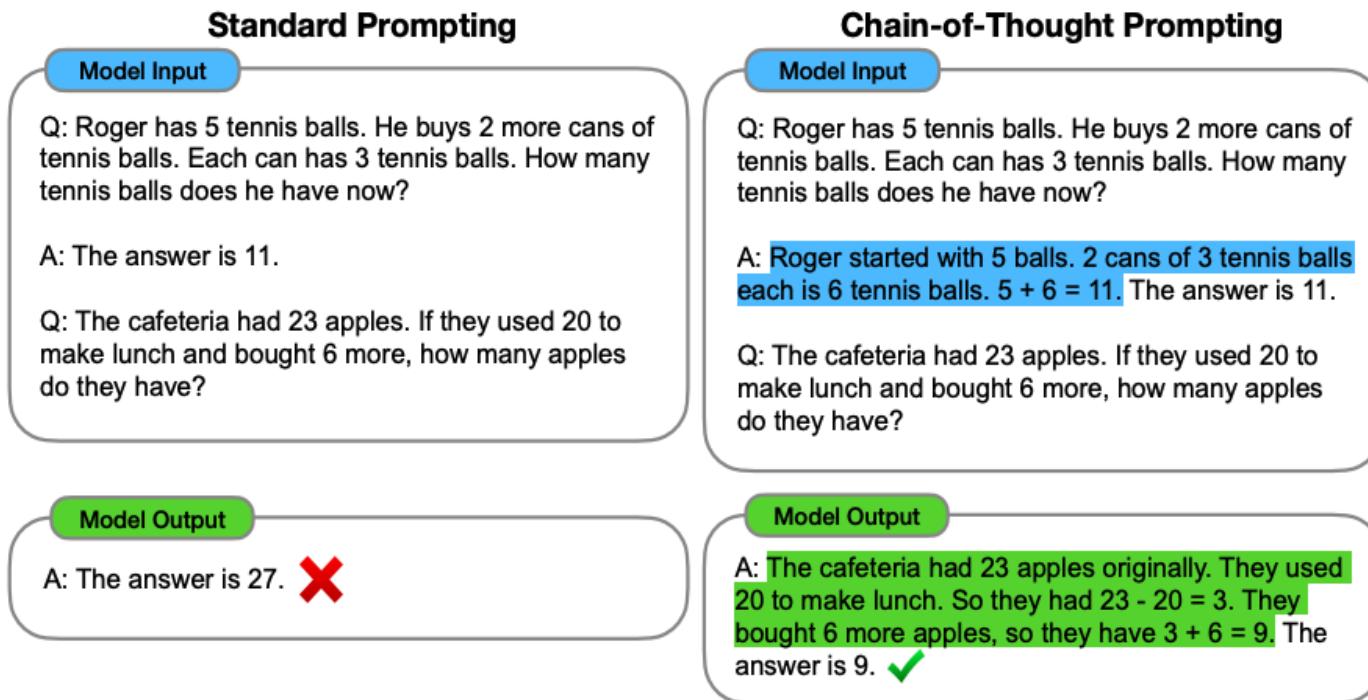
Example 1: I'm sorry to hear that [product name] did not meet your expectations. We strive to ensure every customer is happy with their purchase, and it's clear we have not lived up to this promise in your case. To better assist you, could you please share specific details about how the product did not meet your expectations?

Example 2: I'm so sorry for the delay in getting back to you. I understand your frustration. Your email needed and deserved a timely response. Please contact me directly if you experience any further problems on the site.

A: We're truly sorry to hear about your experience and the frustration it has caused. We value our customers deeply and would like to resolve this promptly. Please email us directly at help@ourcompany.com so we can address your concerns immediately.

Chain-of-thought (CoT) prompting

CoT prompting enables complex reasoning capabilities through intermediate reasoning steps. Can be combined with few-shot prompting to deliver better results.



source: Wei et al. (2022), <https://arxiv.org/abs/2201.11903>

Using prompt schemas

```
import openai

# Define user inputs and campaign details
product_name = "Eco-Friendly Water Bottle"
target_audience = "environmentally conscious consumers"
campaign_theme = "sustainability and durability"
key_features = ["made from 100% recycled materials",
                 "keeps drinks cold for 24 hours"]

# Build a detailed prompt incorporating context, persona, and model role
prompt = f"""
Create a script for a marketing campaign for a product \
called '{product_name}'.
Target audience: {target_audience}.
Campaign theme: {campaign_theme}.
Key features of the product:
- {key_features[0]}
- {key_features[1]}

The script should be engaging and inspire the audience \
to support sustainability by choosing this product. Use \
a friendly and persuasive tone, suitable for social media \
platforms. Include a call to action at the end of the script.
"""

# Call the OpenAI API to generate the script
response = openai.Completion.create(
    engine="text-davinci-003",
    prompt=prompt,
    max_tokens=300,
    temperature=0.7
)
```

Example of a good script:

"Join us in making the earth greener, one sip at a time! \\ Our new {product_name}, designed for {target_audience}, not \\ only looks good but does good. Made entirely from recycled \\ materials, it keeps your drink icy cold for a whole day. \\ Make a difference with every drink. Order yours today and \\ take a step towards a sustainable future!"

Role of the model: You are a creative director who specializes \\ in eco-friendly products and has extensive experience crafting \\ compelling marketing messages for a social media-savvy audience.

```
# Call the OpenAI API to generate the script
response = openai.Completion.create(
    engine="text-davinci-003",
    prompt=prompt,
    max_tokens=300,
    temperature=0.7
)
```

7.4

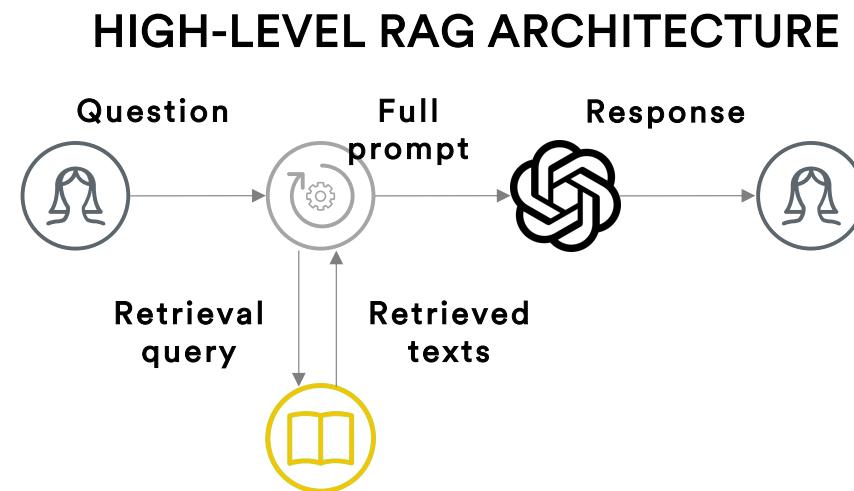
Retrieval-augmented Generation (RAG)

When models access external knowledge
sources

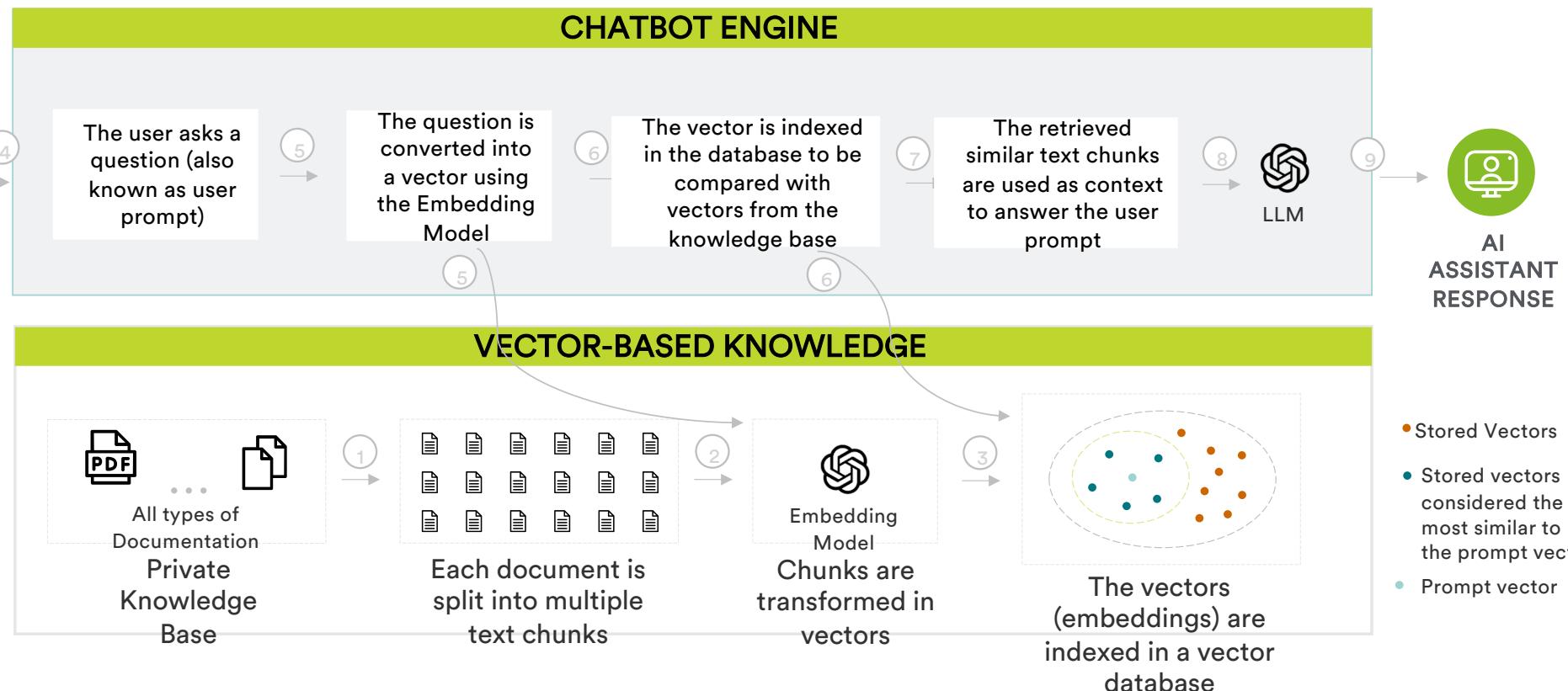
Retrieval-augmented Generation (RAG)

RAG is a technique that combines the generative capabilities of LLMs with the retrieval of external information to produce richer and more accurate responses.

It enhances LLMs' context-awareness by dynamically integrating relevant data during the generation process.



Retrieval-augmented Generation (RAG)





Parameters tuning

Tuning models' outputs through parameter tuning

Simple prompts, return simple answers

The screenshot shows the Azure AI Foundry Chat playground interface. On the left, there's a sidebar with navigation links like Home, Get started, Model catalog, Playgrounds, Chat (selected), Assistants, Audio, Images, Completions, Tools, Fine-tuning, Azure OpenAI Service, Shared resources, Deployments, Quota, Safety + security, Data files, and Vector stores.

The main area is titled "Chat playground" and contains a "Chat history" section with the message "Complete the following sentence. The sky is blue, lemons are yellow and limes are". Below this is a configuration panel with various sliders and input fields:

- Add section:** A button to add a new section.
- Parameters:**
 - Past messages included: 10
 - Max response: 800
 - Temperature: 0.7
 - Top P: 0.95
- Stop sequence:** An input field for stop sequences.
- Frequency penalty:** A slider set to 0.
- Presence penalty:** A slider set to 0.

At the bottom, there's a text input field with placeholder "Type user query here. (Shift + Enter for new line)" and a note "14/128000 tokens to be sent".

Let the model know when to stop

The screenshot shows the Azure AI Foundry Chat playground interface. On the left, there's a sidebar with navigation links like Home, Get started, Model catalog, Playgrounds, Chat (selected), Assistants, Audio, Images, Completions, Tools, Fine-tuning, Azure OpenAI Service Evaluation, Stored completions, Batch jobs, Metrics, Shared resources, Deployments, Quota, Safety + security, Data files, and Vector stores.

The main area is titled "Chat playground". It has sections for "View code", "Deploy", "Import", and "Export". Below these are buttons for "Apply changes" and "Generate prompt".

The "Parameters" section contains the following controls:

- Past messages included: A slider set to 10.
- Max response: A slider set to 5, highlighted with a yellow oval.
- Temperature: A slider set to 0.7.
- Top P: A slider set to 0.95.
- Stop sequence: An input field containing "Stop sequence".
- Frequency penalty: A slider set to 0.
- Presence penalty: A slider set to 0.

The "Chat history" section shows a conversation:

```

...
green.

Complete the following sentence, returning the full sentence. The sky is blue, lemons are yellow and limes are
The sky is blue,

```

A blue box at the bottom is labeled "Type user query here. (Shift + Enter for new line)". At the bottom right, it says "20/128000 tokens to be sent" with icons for microphone and file.

Let the model know when to stop

The screenshot shows the Azure AI Foundry Chat playground interface. On the left, there's a sidebar with navigation links like Home, Get started, Model catalog, Playgrounds, Chat (selected), Assistants, Audio, Images, Completions, Tools, Fine-tuning, Azure OpenAI Service, Evaluation, Stored completions, Batch jobs, Metrics, Shared resources, Deployments, Quota, Safety + security, Data files, and Vector stores.

The main area is titled "Chat playground" and contains a "Chat history" section. A prompt is entered: "Complete the following sentence. The sky is blue, lemons are yellow and limes are". The response starts with "... green." and then continues with "The sky is blue," followed by another continuation prompt. The "Response format" dropdown is set to "Text".

On the right, there are several sliders for generating text:

- Max response: Set to 100 (highlighted with a yellow oval).
- Temperature: Set to 0.7.
- Top P: Set to 0.95.
- Stop sequence: An input field containing "The sky is blue, lemons are yellow, and limes are green." (highlighted with a yellow oval).
- Frequency penalty: Set to 0.
- Presence penalty: Set to 0.

A text input field at the bottom says "Type user query here. (Shift + Enter for new line)".

At the bottom right, it says "37/128000 tokens to be sent".

Temperature controls creativity

The screenshot shows the Azure AI Foundry Chat playground interface. On the left, there's a sidebar with navigation links: Home, Get started, Model catalog, Playgrounds (with Chat selected), Assistants, Audio, Images, Completions, Tools (Fine-tuning, Azure OpenAI Service, Evaluation, Stored completions, Batch jobs, Metrics), Shared resources (Deployments, Quota, Safety + security, Data files, Vector stores), and Vector stores (PREVIEW). The main area is titled "Chat playground" and includes "View code", "Deploy", "Import", and "Export" buttons. It features a "Chat history" section with a message "The sky is blue," followed by a "Parameters" section. The "Temperature" slider is highlighted with a yellow oval and has a value of 0.99. Other sliders include "Max response" (10), "Top P" (0.95), and "Stop sequence". Below these are "Frequency penalty" (0) and "Presence penalty" (0). A text input field at the bottom says "Type user query here. (Shift + Enter for new line)". The top right shows "openaiservice-social (eastus, S0)" and a user profile icon.

Other parameters

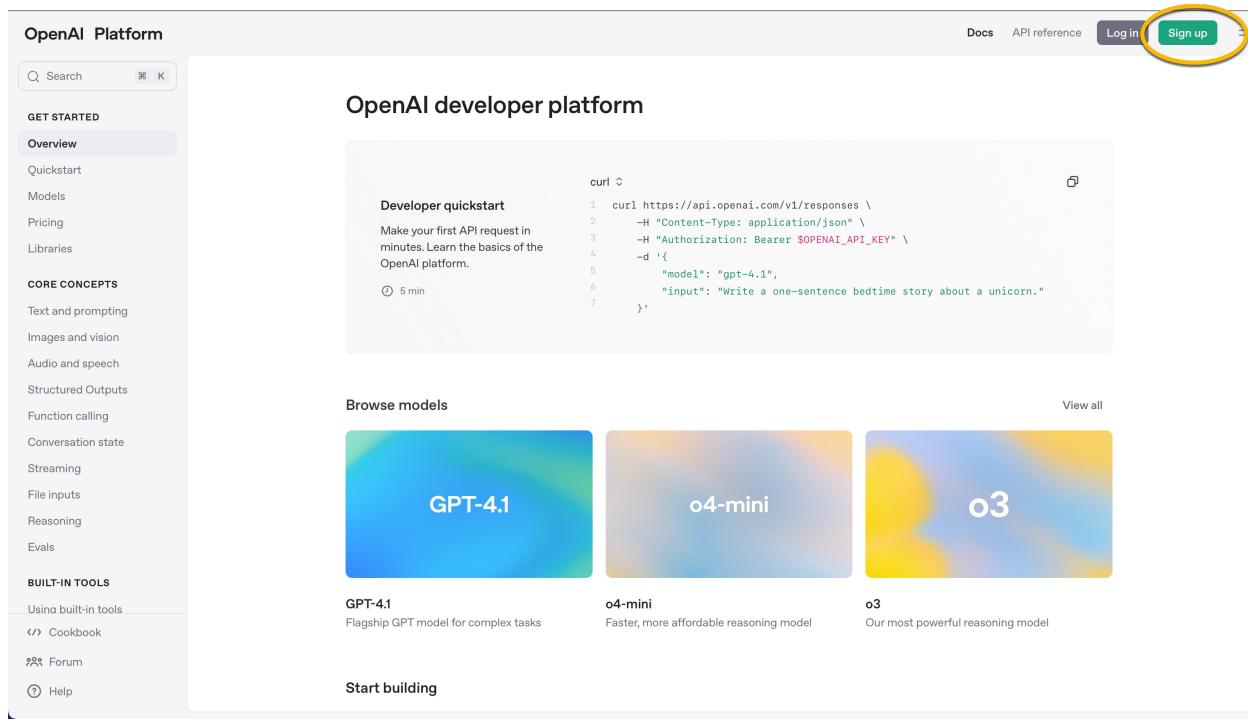
- **Top P:** similar to temperature. Controls randomness but uses a different method. Lowering Top P will narrow the model's token selection to likelier tokens. Increasing Top P will let the model choose from tokens with both high and low likelihood. Try adjusting temperature or Top P but not both
- **Stop sequence:** Make the model end its response at a desired point. The model response will end before the specified sequence, so it won't contain the stop sequence text
- **Frequency penalty:** Reduce the chance of repeating a token proportionally based on how often it has appeared in the text so far. This decreases the likelihood of repeating the exact same text in a response
- **Presence penalty:** Reduce the chance of repeating any token that has appeared in the text at all so far. This increases the likelihood of introducing new topics in a response (good for creativity)

7.6

ChatGPT: Getting an API key

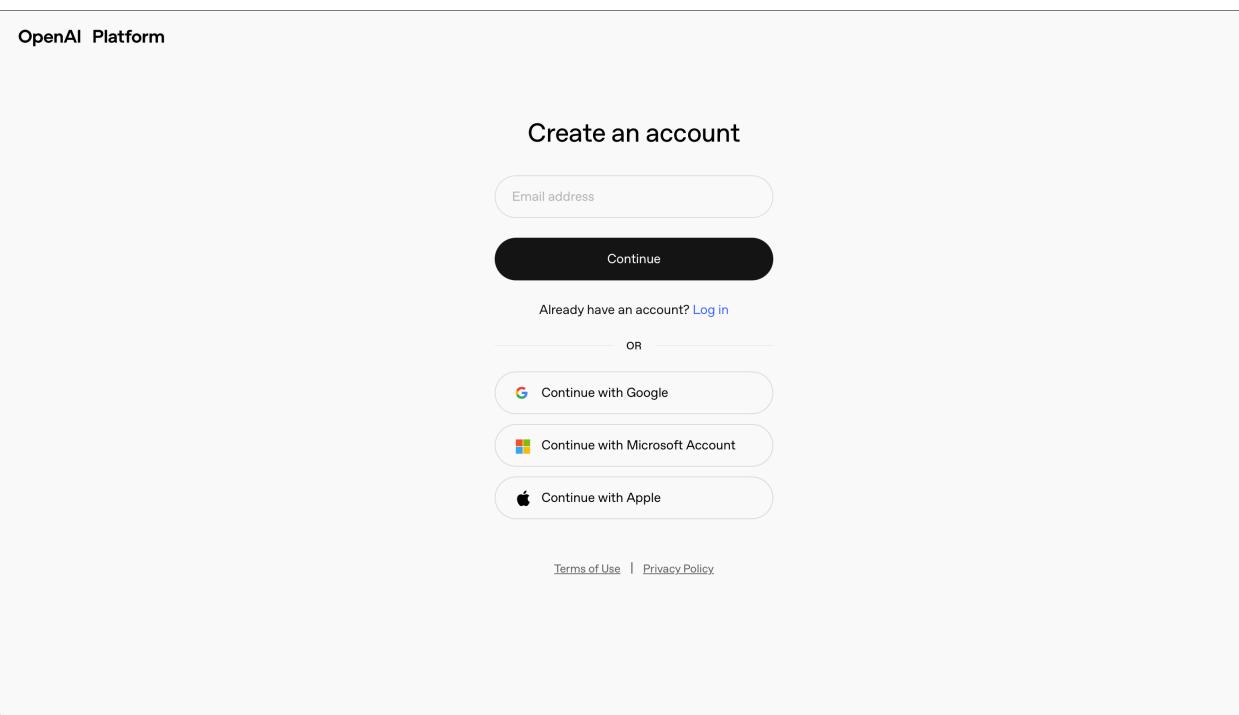
Creating an account in Open AI's platform

- Open a browser and navigate to <https://platform.openai.com>
- Click Sign up



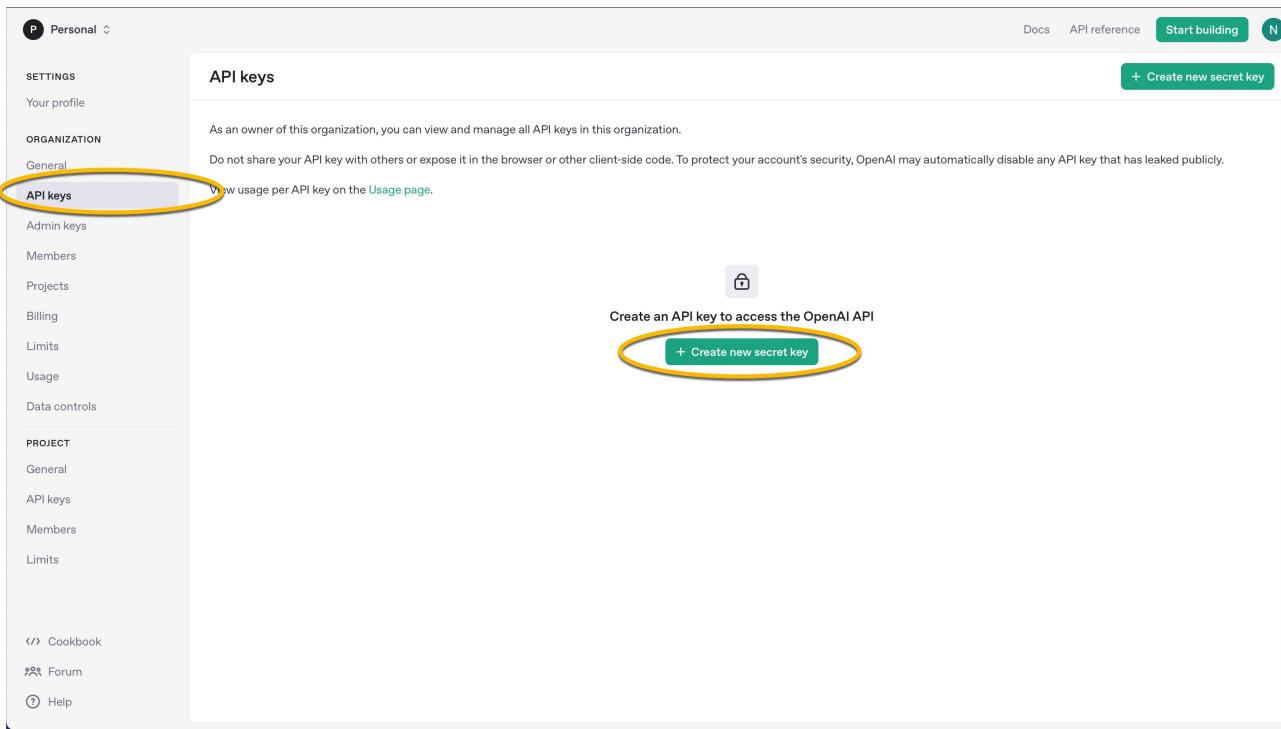
Creating an account in Open AI's platform

- Enter your email address or select an Identity Manager
- Click Continue and follow the steps



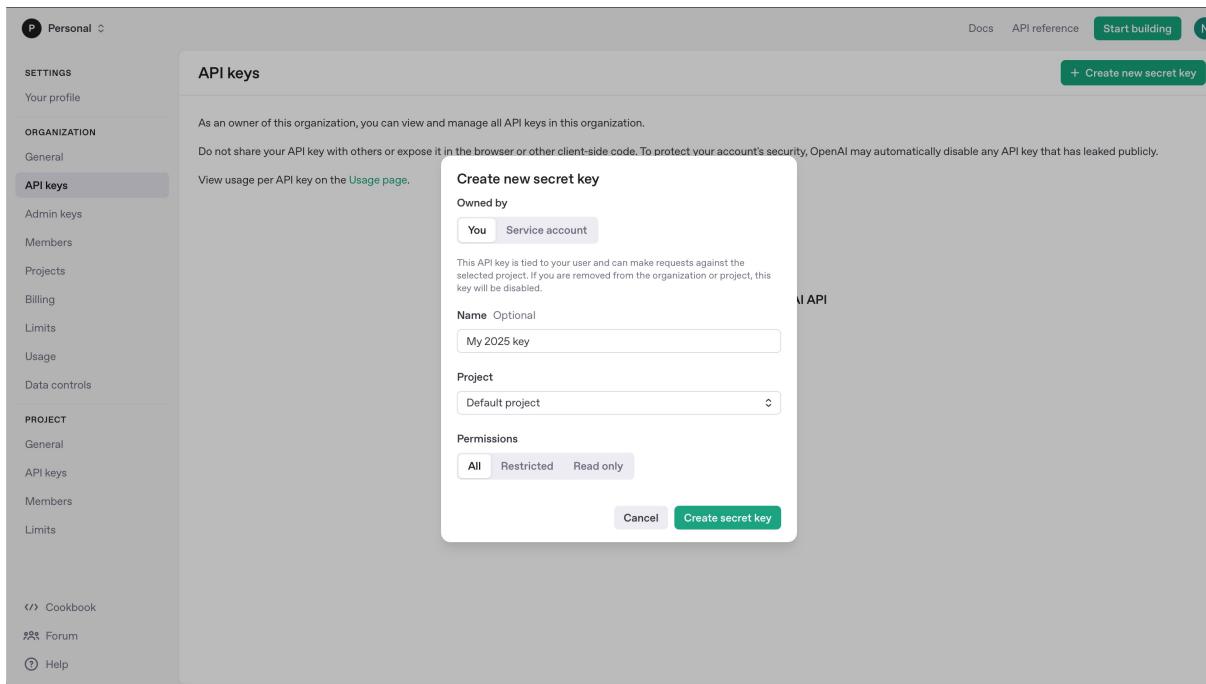
Manage your organization settings and API keys

- Navigate to <https://platform.openai.com/settings/organization/api-keys>
- Click Create new secret key in API keys



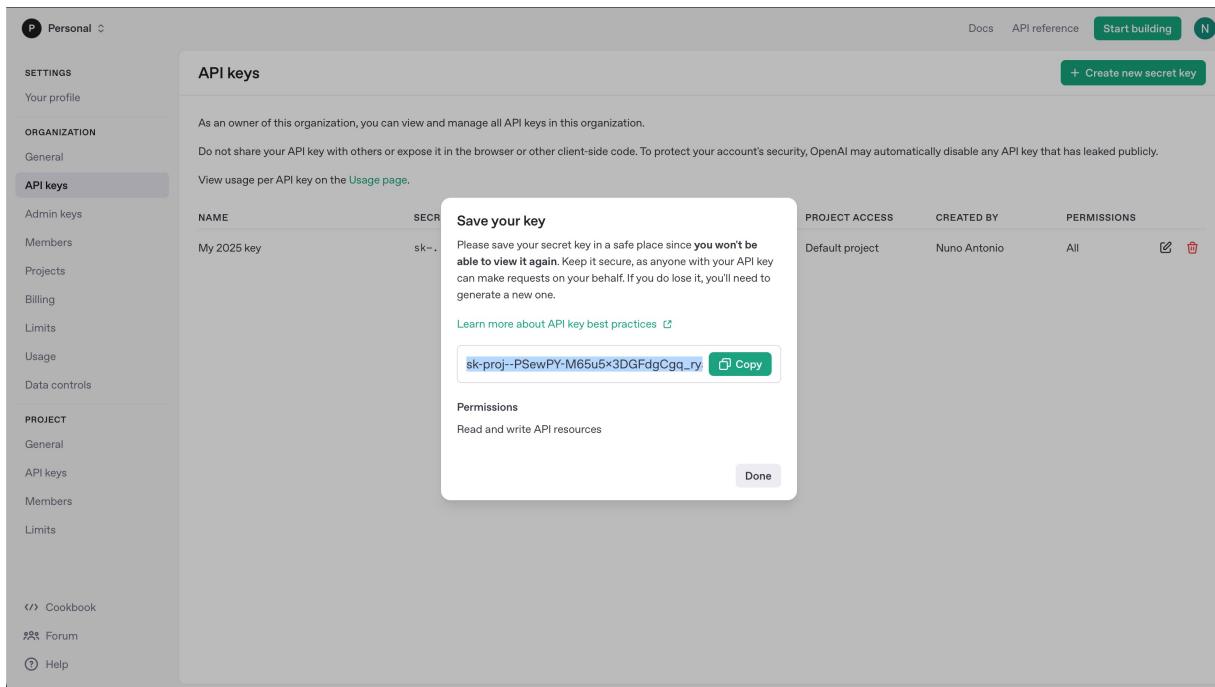
Creating an account in Open AI's platform

- Enter a name for your key (e.g., My Key)
- Click Create new secret key



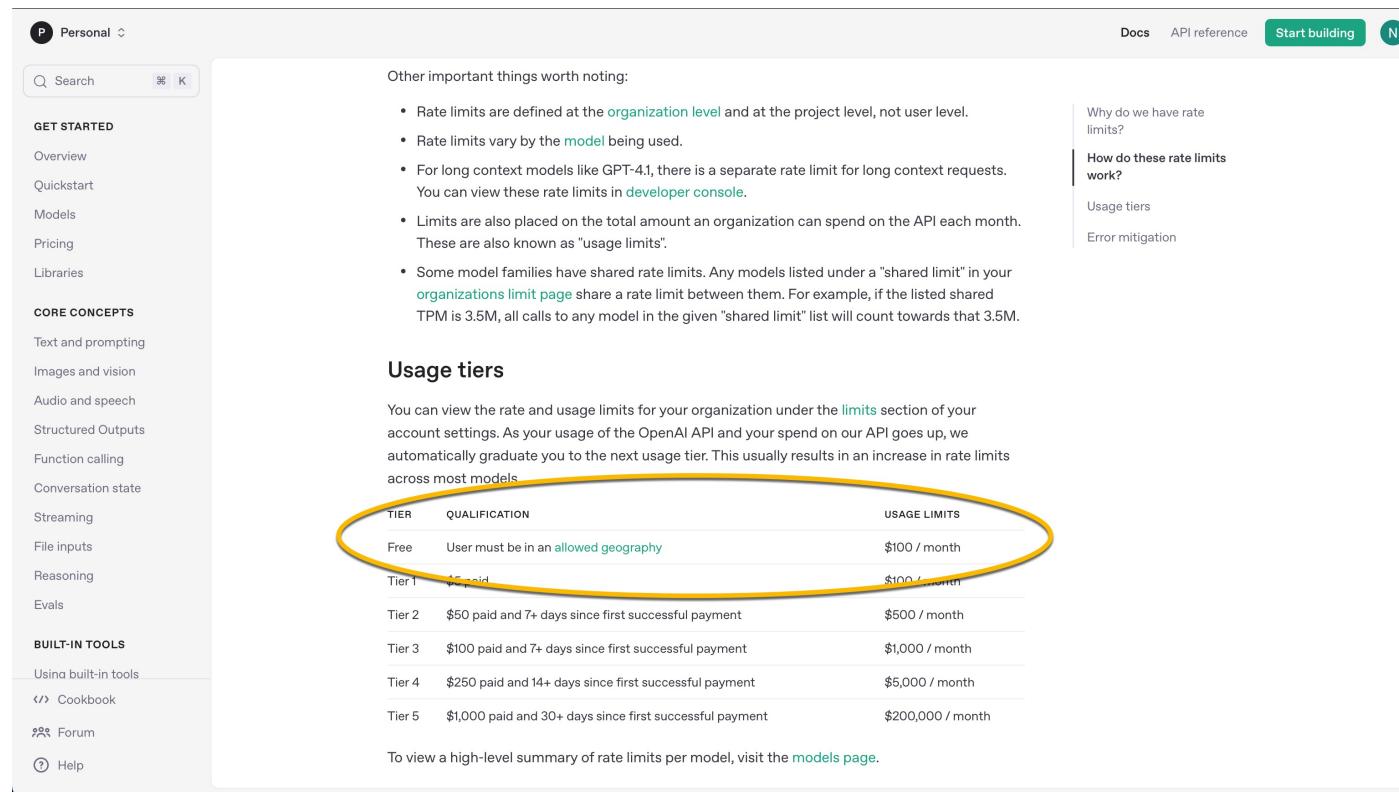
Creating an account in Open AI's platform

- Copy the new key to memory and store the key in a safe place
- Click Done to finish



Creating an account in Open AI's platform

- Navigate to <https://platform.openai.com/docs/guides/rate-limits/usage-tiers?context=tier-free> to check for the Free tier rate limits



The screenshot shows a web browser displaying the OpenAI Platform documentation. The left sidebar has sections like 'GET STARTED' (Overview, Quickstart, Models, Pricing, Libraries), 'CORE CONCEPTS' (Text and prompting, Images and vision, Audio and speech, Structured Outputs, Function calling, Conversation state, Streaming, File inputs, Reasoning, Evals), and 'BUILT-IN TOOLS' (Using built-in tools, Cookbook, Forum, Help). The main content area has a heading 'Other important things worth noting:' followed by a bulleted list about rate limits. Below that is a section titled 'Usage tiers' with a table showing five tiers and their qualifications and usage limits. A yellow oval highlights the 'Free' tier. The table is as follows:

TIER	QUALIFICATION	USAGE LIMITS
Free	User must be in an allowed geography	\$100 / month
Tier 1	\$50 paid	\$100 / month
Tier 2	\$50 paid and 7+ days since first successful payment	\$500 / month
Tier 3	\$100 paid and 7+ days since first successful payment	\$1,000 / month
Tier 4	\$250 paid and 14+ days since first successful payment	\$5,000 / month
Tier 5	\$1,000 paid and 30+ days since first successful payment	\$200,000 / month

At the bottom, it says 'To view a high-level summary of rate limits per model, visit the [models page](#)'.

Pricing

- The cost differs per model

Pricing

Latest models

New: Save on synchronous requests with [flex processing](#).

Model	Input	Cached input	Output
gpt-4.1 ↳ gpt-4.1-2025-04-14	\$2.00	\$0.50	\$8.00
gpt-4.1-mini ↳ gpt-4.1-mini-2025-04-14	\$0.40	\$0.10	\$1.60
gpt-4.1-nano ↳ gpt-4.1-nano-2025-04-14	\$0.10	\$0.025	\$0.40
gpt-4.5-preview ↳ gpt-4.5-preview-2025-02-27	\$75.00	\$37.50	\$150.00
gpt-4o ↳ gpt-4o-2024-08-06	\$2.50	\$1.25	\$10.00
gpt-4o-audio-preview ↳ gpt-4o-audio-preview-2024-12-17	\$2.50	-	\$10.00
gpt-4o-realtime-preview ↳ gpt-4o-realtime-preview-2024-12-17	\$5.00	\$2.50	\$20.00
gpt-4o-mini ↳ gpt-4o-mini-2024-07-18	\$0.15	\$0.075	\$0.60
gpt-4o-mini-audio-preview ↳ gpt-4o-mini-audio-preview-2024-12-17	\$0.15	-	\$0.60
gpt-4o-mini-realtime-preview ↳ gpt-4o-mini-realtime-preview-2024-12-17	\$0.60	\$0.30	\$2.40

Open AI current models (May 2025)

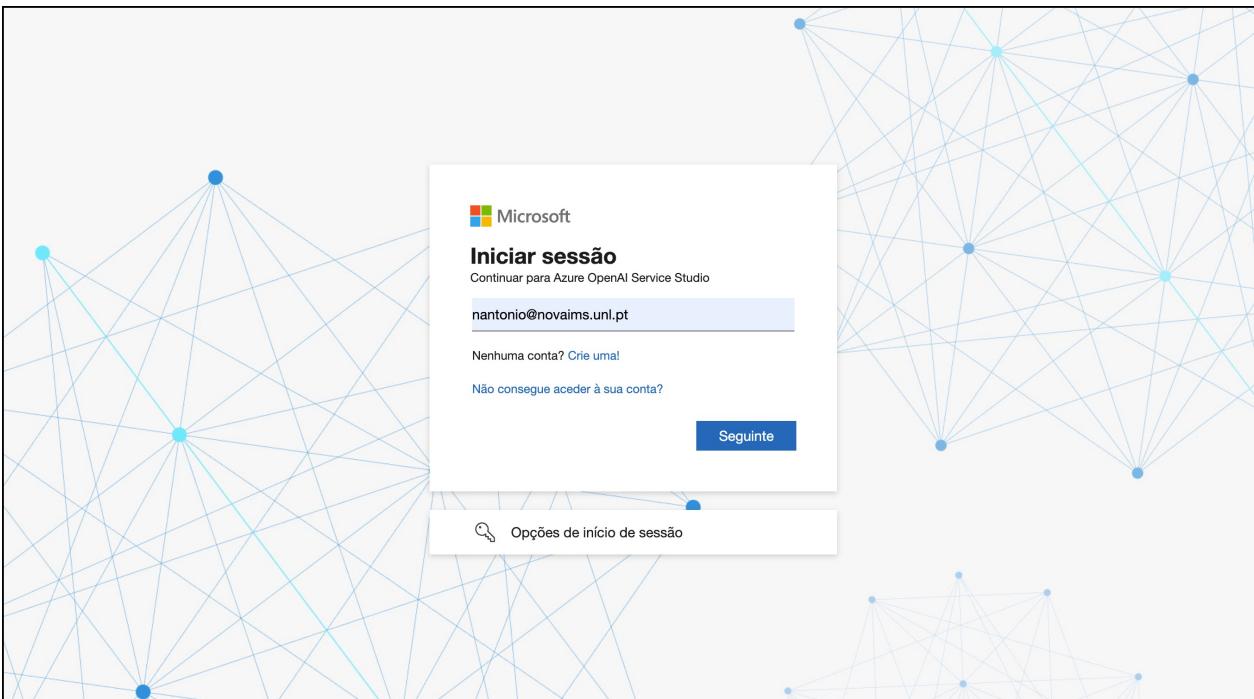
Model	Context window	Description
GPT-4.1	1 047 576 tokens	Flagship multimodal model; unmatched long-document handling, strongest reasoning, robust function/tool calling, excels on text + images.
GPT-4o (Omni)	128 000 tokens	Multimodal, fastest Tier-1 model; top performance in non-English, vision, and code tasks, beating GPT-4 Turbo on speed and accuracy.
o4-mini	200 000 tokens	Reasoning-optimized; superior on math/science, deep logical chains, parallel tool execution, lower cost than GPT-4.1 for complex work.
GPT-4 Turbo (2024-04-09)	128 000 tokens	Production workhorse; balanced price/speed, vision support, JSON mode, reliable for chat, code, and general tasks.
GPT-4o-mini	128 000 tokens	Lightweight Omni variant; multimodal, very fast and inexpensive—ideal drop-in upgrade for GPT-3.5-scale workloads.

7.7

Azure Open AI

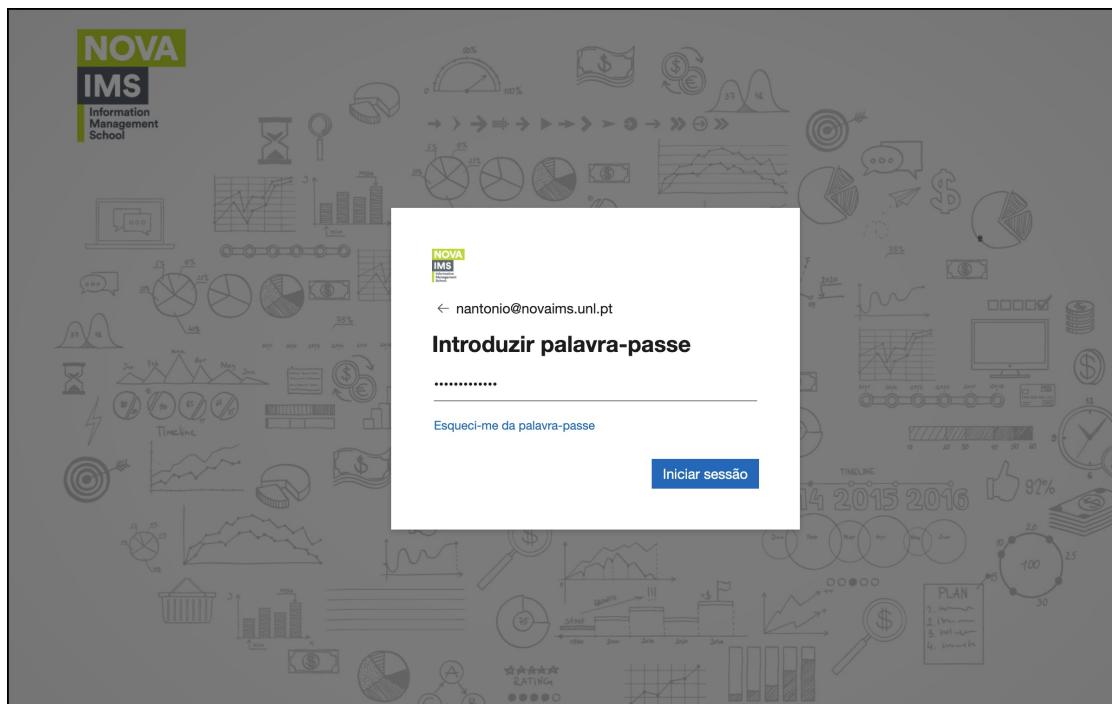
Using NOVA IMS' Azure Open AI service

- Open a browser and navigate to <https://oai.azure.com/portal>
- Enter you NOVA IMS email address and click Next



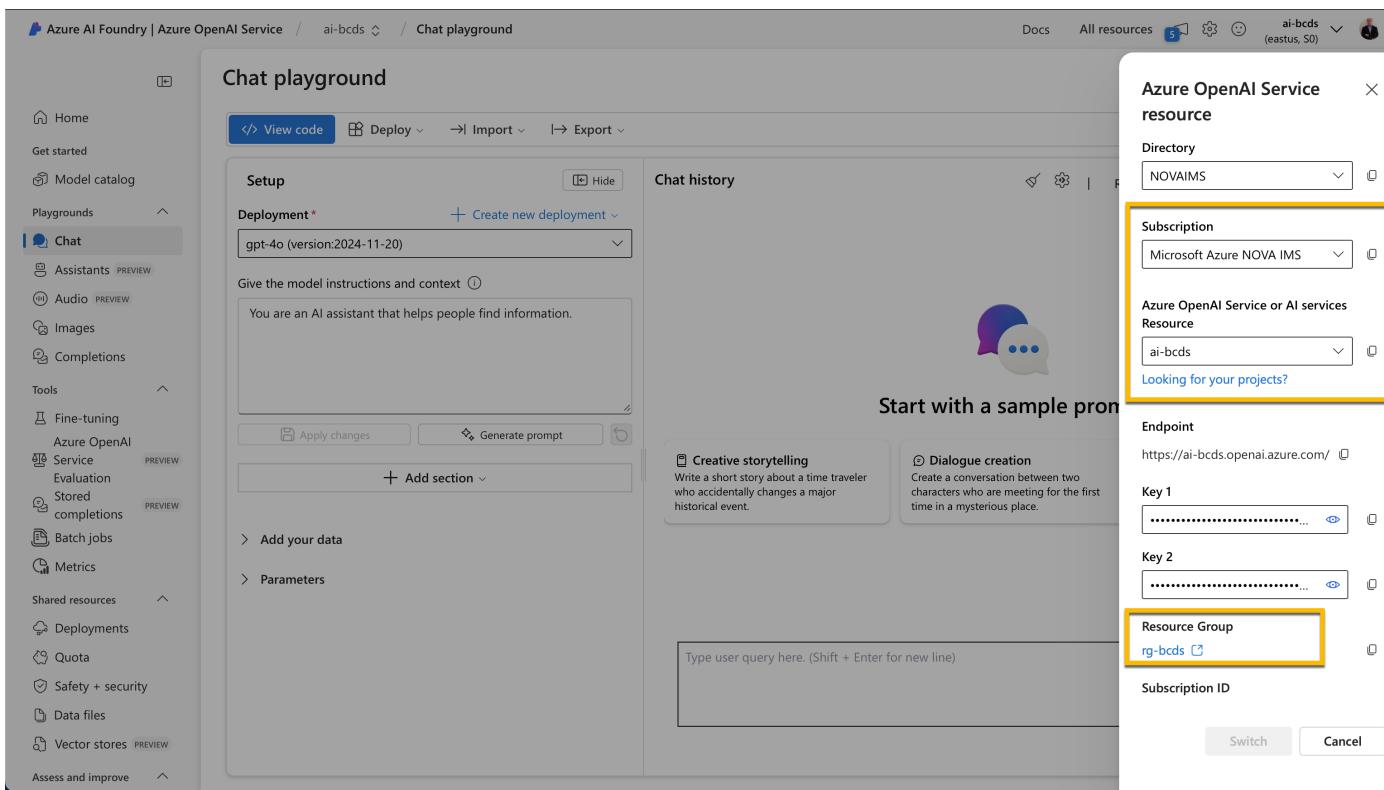
Using NOVA IMS' Azure Open AI service

- Enter your NOVA IMS password and click Next
- In the following window, click to maintain the session opened



Select the resource for classes

- Select the Subscription “Microsoft Azure NOVA IMS” and Resource “ai-bcds” (the Resource Group is “rg-bcds”)



Azure Open AI

Explore the functionalities

The screenshot shows the Azure OpenAI Service interface. On the left, there is a sidebar with various navigation options: Home, Get started, Model catalog, Playgrounds, Chat, Assistants (PREVIEW), Audio (PREVIEW), Images, Completions, Tools, Fine-tuning, Azure OpenAI Service (PREVIEW), Shared resources, Deployments, Quota, Safety + security, Data files, and Vector stores (PREVIEW). The main content area is titled "Welcome to Azure OpenAI Service" and says "Explore the generative AI models and craft unique prompts for your use cases." It displays "Resource configuration" details: Name (openaiservice-social), Subscription (Microsoft Azure NOVA IMS), Subscription ID (111188e6-996f-4f86-9280-43b0dd363e4a), API key 1, Resource group (RG_SOCIAL_ANALYTICS), Pricing tier (Standard S0), API key 2, Azure OpenAI Service endpoint (https://openaiservice-social.openai.azure.com/), and Location (eastus). Below this, there are six "Get started" cards: Assistants playground, Chat playground, Bring your own data, Completions playground, Images playground, and Fine-tuning.

Use the Chat Playground to use GPT 4o

Azure AI Foundry | Azure OpenAI Service / openaiservice-social / Chat playground

Switch to the old look Docs All resources openaiservice-social (eastus, S0)

Chat playground

View code Deploy Import Export

Setup Deployment * + Create new deployment

4o (version:2024-08-06) Hide

Please select the 'Apply changes' button below.

Give me model instructions and context

You are an AI assistant who specializes in creating posts for social media platforms. All the posts should sound professional, based on fact, and never exceed 200 words.

Apply changes Generate prompt

+ Add section

Chat history

Write a LinkedIn post that announces that applications for the NOVA IMS's Data-Driven Marketing Master will open on June 1st

Exciting News for Aspiring Data-Driven Marketers! 🌟

We are thrilled to announce that applications for the highly anticipated NOVA IMS Data-Driven Marketing Master program will officially open on June 1st! 🎉

Are you ready to elevate your marketing skills with cutting-edge data analytics and strategic insights? Our program offers a unique blend of academic excellence and real-world expertise, designed to empower you in the ever-evolving marketing landscape.

What to Expect:

- ✓ Comprehensive curriculum covering the latest in data-driven marketing techniques
- ✓ Hands-on experience with industry-leading tools and technologies
- ✓ Networking opportunities with top professionals and thought leaders
- ✓ Personalized career support to help you achieve your professional goals

Whether you're looking to advance your career, pivot into the dynamic world of data-driven marketing, or simply expand your knowledge, this program is your gateway to success.

Mark your calendars and be ready to apply starting June 1st! Stay tuned for more details and insights into what makes our program the perfect fit for your aspirations.

#DataDrivenMarketing #NOVAIMS #MasterProgram #MarketingInnovation #CareerGrowth #ApplyNow

If you have any questions or need further information, feel free to reach out. We can't wait to welcome the next cohort of future marketing leaders! 🚀

Type user query here. (Shift + Enter for new line)

289/128000 tokens to be sent

You can add your own data

The screenshot shows the Azure AI Foundry Chat playground interface. On the left, there's a sidebar with navigation links like Home, Get started, Model catalog, Playgrounds, Chat (which is selected), Assistants, Audio, Images, Completions, Tools, Fine-tuning, Service Evaluation, Stored completions, Batch jobs, Metrics, Shared resources, Deployments, Quota, Safety + security, Data files, and Vector stores. The main area has tabs for View code, Deploy, Import, and Export. The 'Setup' section includes a Deployment dropdown set to '4o (version:2024-08-06)', a 'Give the model instructions and context' text area, and buttons for Apply changes, Generate prompt, and Add section. Below this, there are two buttons circled in yellow: '+ Add your data' and '+ Add a data source'. The 'Chat history' section contains a message about the NOVA IMS Data-Driven Marketing Master program opening on June 1st, followed by a list of what to expect, a summary of the program, and a conclusion. At the bottom, there's a 'New chat session started' message, a note about previous messages not being used as context, and a text input field for user queries.

Azure AI Foundry | Azure OpenAI Service / openaiservice-social / Chat playground

Switch to the old look Docs All resources openaiservice-social (eastus, S0)

Help

Chat playground

View code Deploy Import Export

Setup Hide

Deployment * Create new deployment

4o (version:2024-08-06)

Give the model instructions and context

You are an AI assistant who specializes in creating posts for social media platforms. All the posts should sound professional, based on fact, and never exceed 200 words.

Apply changes Generate prompt Add section

+ Add your data

+ Add a data source

Parameters

Chat history

Write a LinkedIn post that announces that applications for the NOVA IMS's Data-Driven Marketing Master will open on June 1st

Exciting News for Aspiring Data-Driven Marketers!

We are thrilled to announce that applications for the highly anticipated NOVA IMS Data-Driven Marketing Master program will officially open on June 1st! 🎉

Are you ready to elevate your marketing skills with cutting-edge data analytics and strategic insights? Our program offers a unique blend of academic excellence and real-world expertise, designed to empower you in the ever-evolving marketing landscape.

What to Expect:

- Comprehensive curriculum covering the latest in data-driven marketing techniques
- Hands-on experience with industry-leading tools and technologies
- Networking opportunities with top professionals and thought leaders
- Personalized career support to help you achieve your professional goals

Whether you're looking to advance your career, pivot into the dynamic world of data-driven marketing, or simply expand your knowledge, this program is your gateway to success.

Mark your calendars and be ready to apply starting June 1st! Stay tuned for more details and insights into what makes our program the perfect fit for your aspirations.

#DataDrivenMarketing #NOVAIMS #MasterProgram #MarketingInnovation #CareerGrowth #ApplyNow

If you have any questions or need further information, feel free to reach out. We can't wait to welcome the next cohort of future marketing leaders! 🎉

New chat session started

The assistant setup has been updated. Previous messages won't be used as context for new queries.

Type user query here. (Shift + Enter for new line)

33/128000 tokens to be sent

Tune parameters

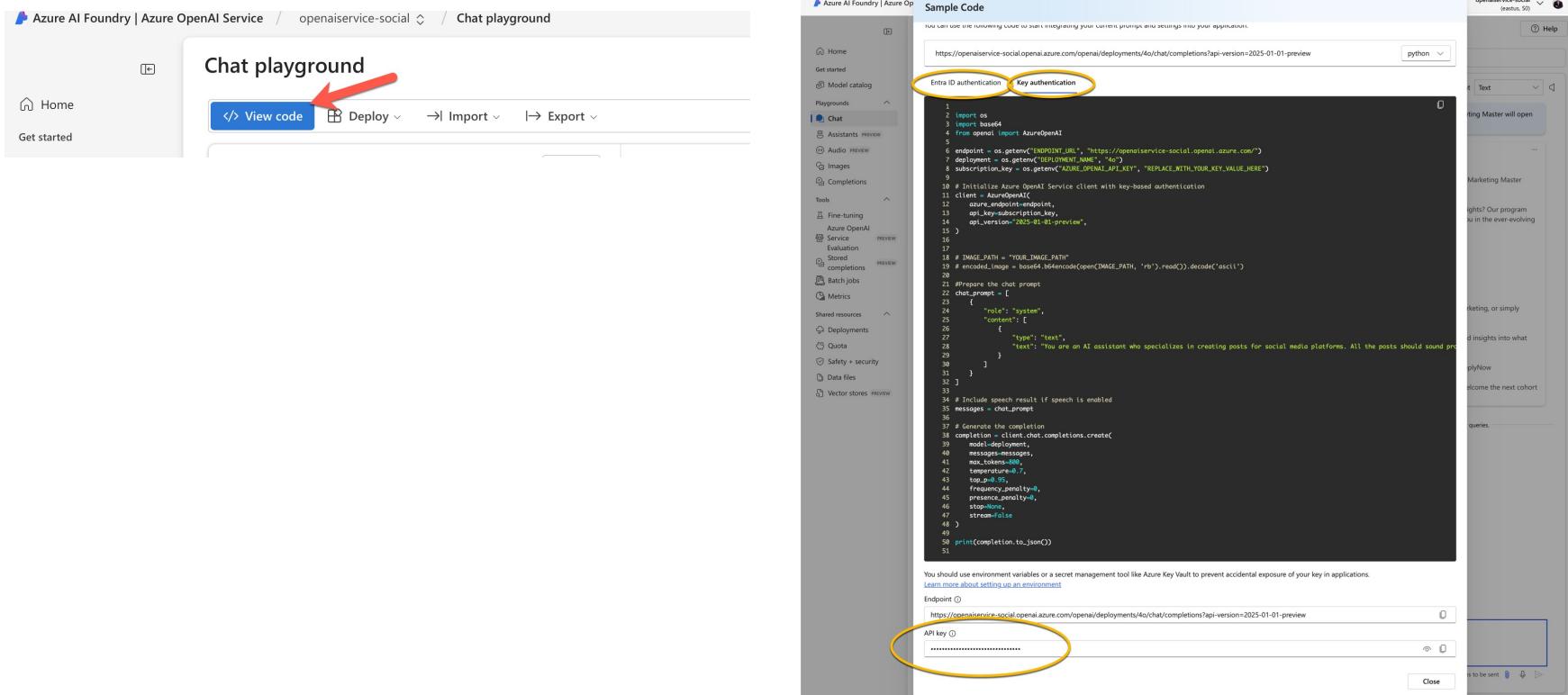
The screenshot shows the Azure AI Foundry Chat playground interface. On the left, there's a sidebar with navigation links like Home, Get started, Model catalog, Playgrounds, Chat (selected), Assistants, Audio, Images, Completions, Tools, Fine-tuning, Service Evaluation, Stored completions, Batch jobs, Metrics, Shared resources, Deployments, Quota, Safety + security, Data files, and Vector stores. The Chat section has sub-links for Assistants, Audio, Images, and Completions.

The main area is titled "Chat playground". It has a "Give the model instructions and context" section containing a text input field with the placeholder: "You are an AI assistant who specializes in creating posts for social media platforms. All the posts should sound professional, based on fact, and never exceed 200 words." Below this is a "Parameters" section, which is circled in yellow. This section contains sliders for "Past messages included" (set to 10), "Max response" (set to 800), "Temperature" (set to 0.7), "Top P" (set to 0.95), and "Stop sequence" (empty). It also includes "Frequency penalty" (set to 0) and "Presence penalty" (set to 0).

To the right is the "Chat history" section. It shows a message from the user: "Write a LinkedIn post that announces that applications for the NOVA IMS's Data-Driven Marketing Master will open on June 1st". The AI's response is: "🌟 Exciting News for Aspiring Data-Driven Marketers! 🌟 We are thrilled to announce that applications for the highly anticipated NOVA IMS Data-Driven Marketing Master program will officially open on June 1st! 🎉 Are you ready to elevate your marketing skills with cutting-edge data analytics and strategic insights? Our program offers a unique blend of academic excellence and real-world expertise, designed to empower you in the ever-evolving marketing landscape." Below this, there's a "What to Expect:" section with a bulleted list of benefits, and a "Mark your calendars" section with a note about the start date and hashtags: "#DataDrivenMarketing #NOVAIMS #MasterProgram #MarketingInnovation #CareerGrowth #ApplyNow". At the bottom, there's a "New chat session started" message and a note about token usage: "The assistant setup has been updated. Previous messages won't be used as context for new queries." A text input field at the bottom says "Type user query here. (Shift + Enter for new line)".

Obtaining the Python code

- Click the **View code** to check the Python code to call
 - Get your API token in the link bellow the code
 - You can copy the code by clicking the **Copy** button



References

- <https://github.com/Mooler0410/LLMsPracticalGuide>
- <https://www.slideshare.net/rmcdermo/fhllmtalk20231005pdf>
- <https://learn.microsoft.com/en-us/training/modules/introduction-large-language-models/2-understand-large-language-models>
- <https://developer.nvidia.com/blog/how-to-get-better-outputs-from-your-large-language-model/>
- https://en.wikipedia.org/wiki/Prompt_engineering
- <https://learn.microsoft.com/en-us/azure/ai-services/openai/reference>

Business Cases with Data Science

© 2020-2025 Nuno António and Hugo Silva (rev. 2025-05-04)

Acreditações e Certificações



Instituto Superior de Estatística e Gestão da Informação
Universidade Nova de Lisboa