

Hugging Face and open-source generative AI

Niels Rogge



Hi, I'm Niels!

Graduated as Business and Informations Systems Engineer

KU LEUVEN

Machine Learning Engineer at Hugging Face & ML6



ML6

Contributing to open-source (Transformers library)

Implementing AI use cases end-to-end for clients in the Benelux (mainly generative AI)

Making tutorials for people (Transformers-Tutorials, YouTube)



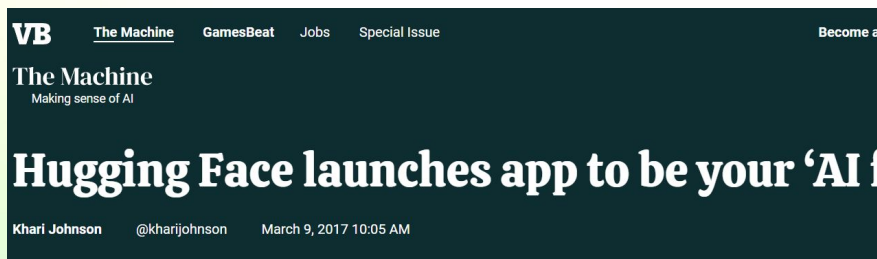
Content

1. What is Hugging Face
2. Hugging Face ecosystem
3. The rise of (open) LLMs
4. Current trends
5. Business cases



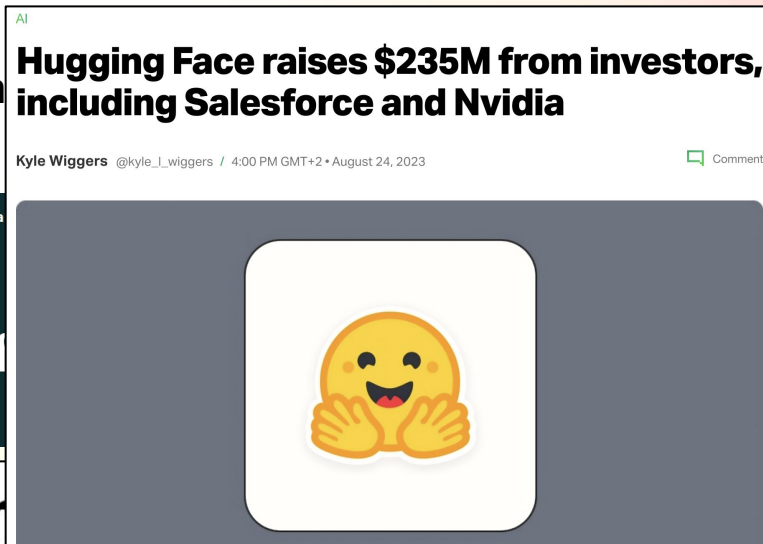
What is Hugging Face

- Founded in 2016
- Started as a chatbot company...
- ... but pivoted to **open-source** and **open**
- 250 employees



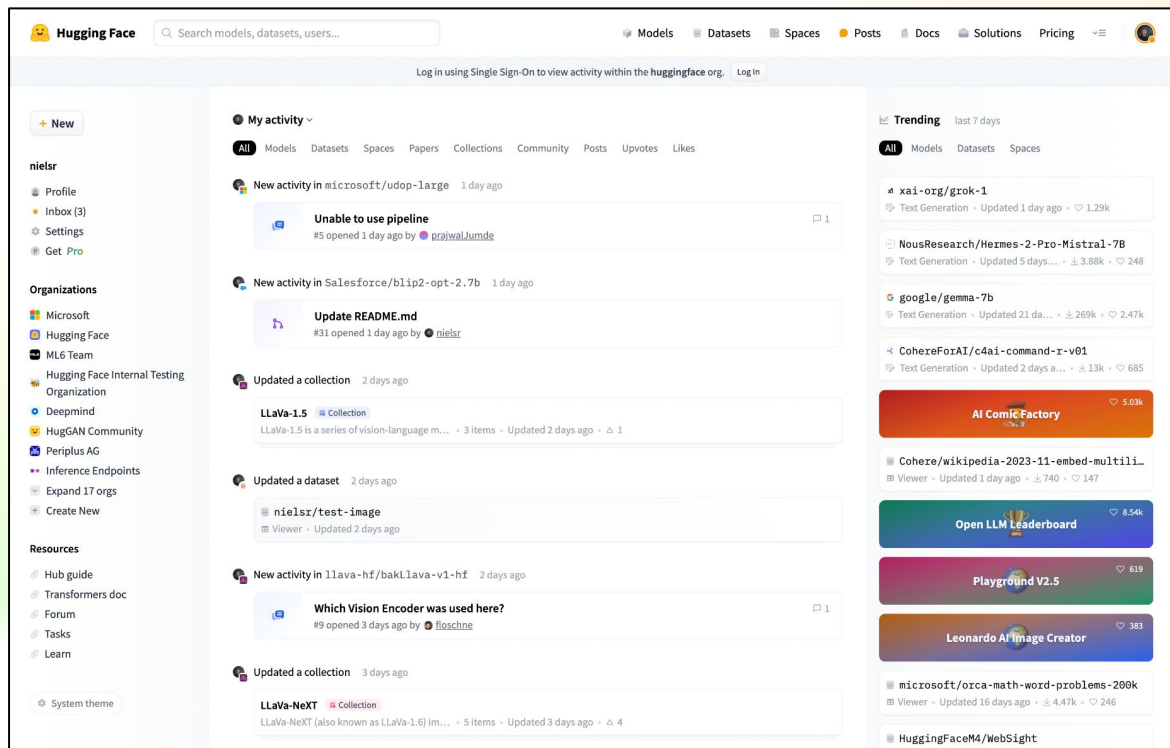
Hugging Face raises \$40 m natural language processing library

Romain Dillet @romaindillet / 4:11 PM GMT+1 • March 11, 2021



What is Hugging Face

- Home of open, **collaborative** machine learning (“Github of AI/ML”)



The screenshot displays the Hugging Face website interface. At the top, there is a navigation bar with the Hugging Face logo, a search bar, and links for Models, Datasets, Spaces, Posts, Docs, Solutions, and Pricing. Below the navigation bar, a user is logged in using Single Sign-On. The main content area is divided into three columns. The left column contains a sidebar with a '+ New' button, a user profile for 'nielsr', and a list of organizations including Microsoft, Hugging Face, and ML6 Team. The middle column shows a 'My activity' feed with several items: a message 'Unable to use pipeline', an update to a README file, a collection update for 'LLaVa-1.5', a dataset update for 'nielsr/test-image', and a collection update for 'LLaVa-NeXT'. The right column features a 'Trending' section with a list of popular models and datasets, including 'xai-org/grok-1', 'NousResearch/Heimes-2-Pro-Mistral-7B', 'google/gemma-7b', 'AI Comic Factory', 'Open LLM Leaderboard', 'Playground V2.5', and 'Leonardo AI Image Creator'.



Used everywhere in the AI world

15,000+ startups and enterprises



Open source contributors



Hardware partners



Hugging Face

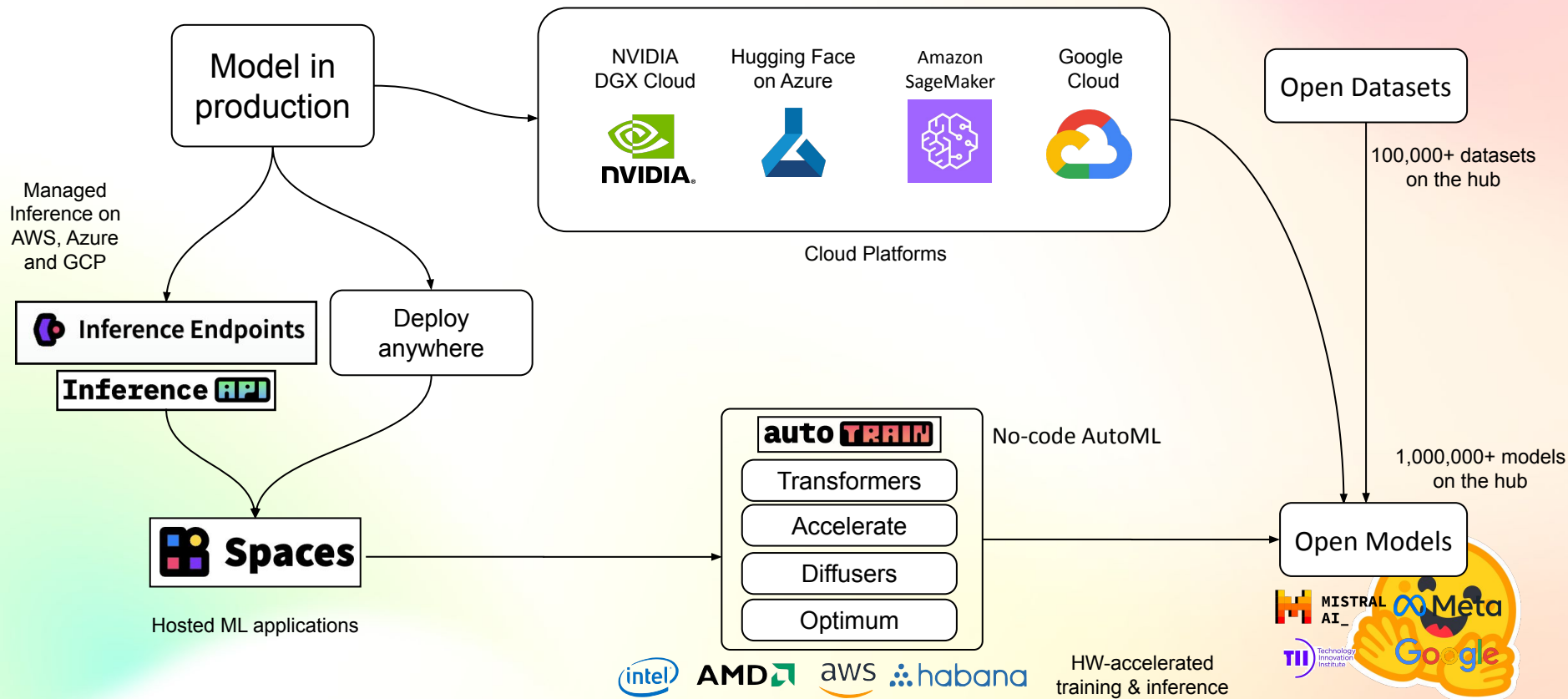
Cloud partners



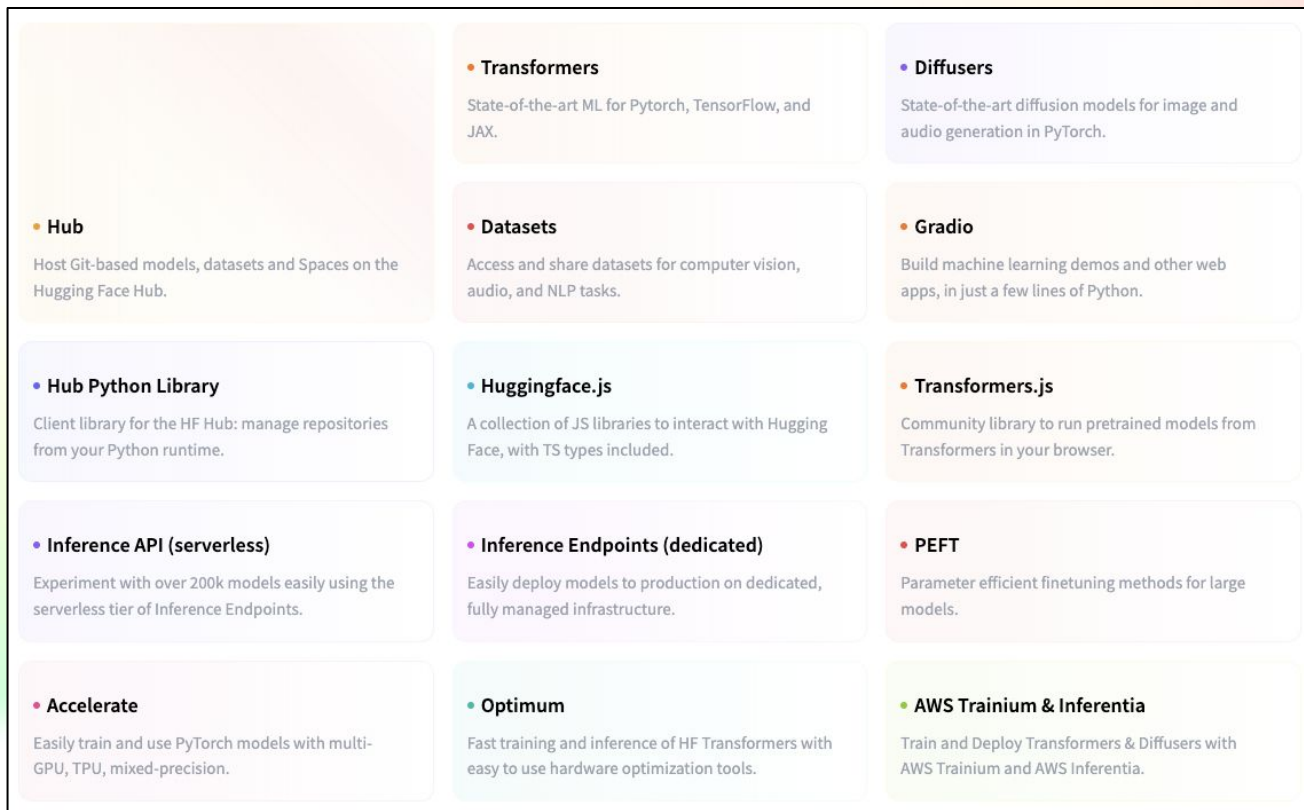
On-prem partners



Integrated with the ML ecosystem



Hugging Face ecosystem



The rise of (open) LLMs

2018: BERT

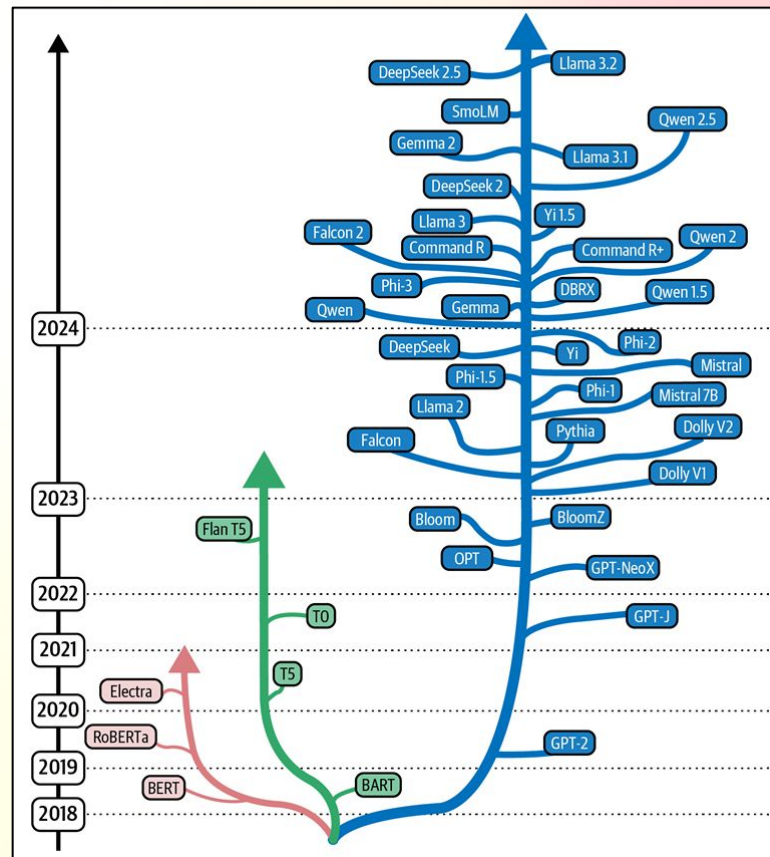
2019: RoBERTa, GPT-2

2020: T5

(...)

2024: open LLMs from

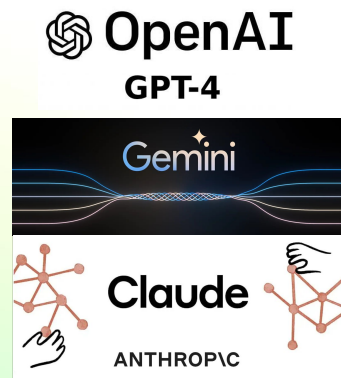
- Meta
- Google
- Cohere
- Mistral
- Alibaba
- ...



The rise of (open) LLMs

LLM families

Closed models



model weights not available

- can't run the model locally
- no access to model's internals
- limits fine-tuning abilities

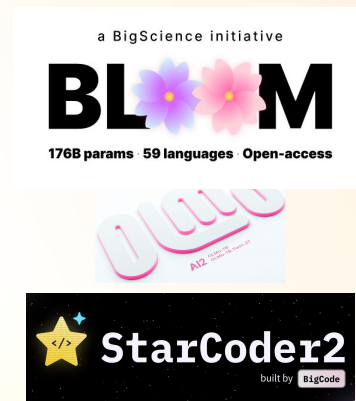
Open models



no access to training data or code

- who's data is in the dataset?
- benchmark contamination
- limits scientific reproducibility

Fully open models




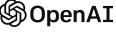







full access to model/code/data

- competitive edge
- liability issues
- maintenance



The rise of (open) LLMs

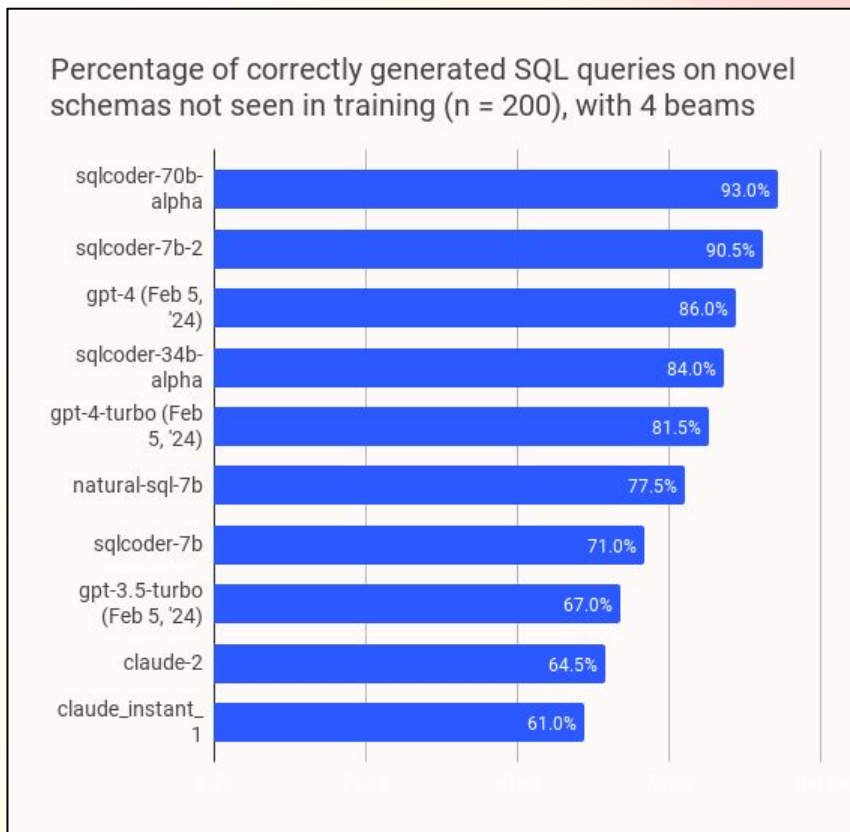
Open vs. closed

| | Open-Source | Closed / Proprietary |
|---------------|---|---|
| Security | Models can be self-hosted , data stays in your environment | Models cannot be self-hosted. Data is sent outside your environment to vendor |
| Control | The timing and nature of updates are controlled by you | Updates and changes to performance can happen without notice |
| Customization | Full source code access to customize the model for your needs | Limited ability to customize for your needs |
| Transparency | Inspect code and data provides better auditability and understandability | No ability to audit or understand performance |
| Cost | Typical lower long term cost due to smaller model size | larger model size and proprietary premium often balanced by decreased cost from server-side optimization |
| Latency | Lower latency due to on premise and smaller model sizes | Often greater latency due to larger model sizes + API latency |
| Quality | No single approach is best. Each use case will vary. Proprietary is typically closer to the frontier of performance . | |
| Examples |   OpenAI  Meta  salesforce  FLAN-T5  MISTRAL AI  Microsoft |  OpenAI  ANTHROPIC |

The rise of (open) LLMs

Pro of open-source: easy to fine-tune


- oftentimes **boosts performance** on particular domain/task
- E.g. [text-to-SQL](#)



The rise of (open) LLMs

Pro of open-source: easy to fine-tune

- oftentimes **boosts performance** on particular domain/task
- E.g. [function calling](#)

 **Berkeley Function-Calling Leaderboard**

BFCL Leaderboard

The Berkeley Function Calling Leaderboard V2 (also called Berkeley Tool Calling Leaderboard V2) evaluates the LLM's ability to call functions (aka tools) accurately. This leaderboard consists of real-world data and will be updated periodically. For more information on the evaluation dataset and methodology, please refer to our [blog post](#) and [code release](#).

Last Updated: 2024-09-13 [[Change Log](#)]

[Expand/Collapse Table](#) [See Only Live Category](#)

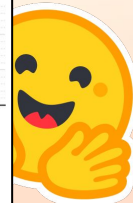
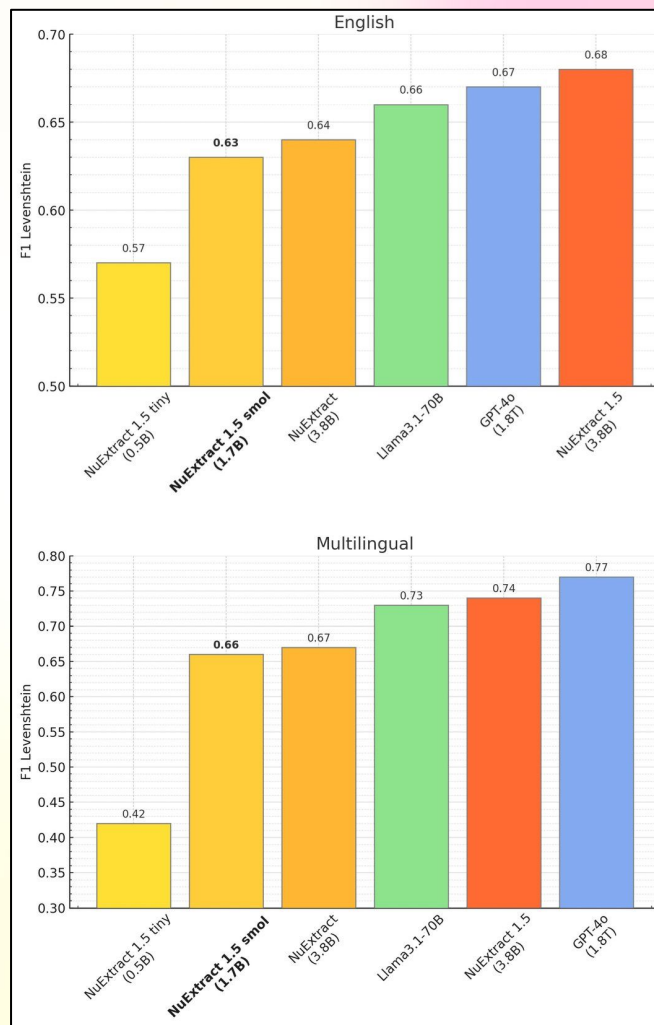
| Rank | Overall Acc | Model | Cost (\$) | Average Latency (s) | AST Summary | Exec Summary | Irrelevance | Relevance | Organization | License |
|------|-------------|------------------------------------|-----------|---------------------|-------------|--------------|-------------|-----------|--------------|------------------------|
| 1 | 85.89 | xLAM-8x22b-r (FC) | N/A | N/A | 82.21 | 89.32 | 75.23 | 97.56 | Salesforce | cc-by-nc-4.0 |
| 2 | 85.79 | GPT-4-0125-Preview (Prompt) | 7.76 | 1.46 | 85.5 | 89.25 | 61.35 | 97.56 | OpenAI | Proprietary |
| 3 | 85 | GPT-4-1106-Preview (Prompt) | 7.88 | 1.58 | 86.31 | 87.38 | 64.98 | 90.24 | OpenAI | Proprietary |
| 4 | 84.74 | GPT-4-0613 (Prompt) | 22.73 | 1.67 | 84.66 | 87.57 | 75.57 | 82.93 | OpenAI | Proprietary |
| 5 | 83.89 | GPT-4-turbo-2024-04-09 (Prompt) | 7.81 | 1.4 | 85.41 | 88.12 | 61.82 | 82.93 | OpenAI | Proprietary |
| 6 | 83.35 | GPT-4o-mini-2024-07-18 (Prompt) | 0.12 | 0.62 | 80.51 | 87.95 | 79.2 | 80.49 | OpenAI | Proprietary |
| 7 | 83.13 | GPT-4o-2024-05-13 (Prompt) | 3.92 | 0.71 | 83.83 | 85.12 | 77.44 | 78.05 | OpenAI | Proprietary |
| 8 | 82.58 | Hammer-7b (FC) | N/A | N/A | 77.64 | 87.39 | 72.98 | 92.68 | MadeAgents | cc-by-nc-4.0 |
| 9 | 82.55 | Functionary-Medium-v3.1 (FC) | N/A | 5.06 | 81.06 | 89.32 | 73.23 | 70.73 | MeetKai | MIT |
| 10 | 82.06 | xLAM-8x7b-r (FC) | N/A | N/A | 77.99 | 85.89 | 72.35 | 92.68 | Salesforce | cc-by-nc-4.0 |
| 11 | 81.78 | GPT-4-1106-Preview (FC) | 6.72 | 3.46 | 77.95 | 87.61 | 72.7 | 82.93 | OpenAI | Proprietary |
| 12 | 81.21 | Meta-Llama-3-70B-Instruct (Prompt) | 1.1 | 0.18 | 79.79 | 87.41 | 50.63 | 92.68 | Meta | Meta Llama 3 Community |

FC = native support for function/tool calling.

The rise of (open) LLMs

Pro of open-source: easy to fine-tune

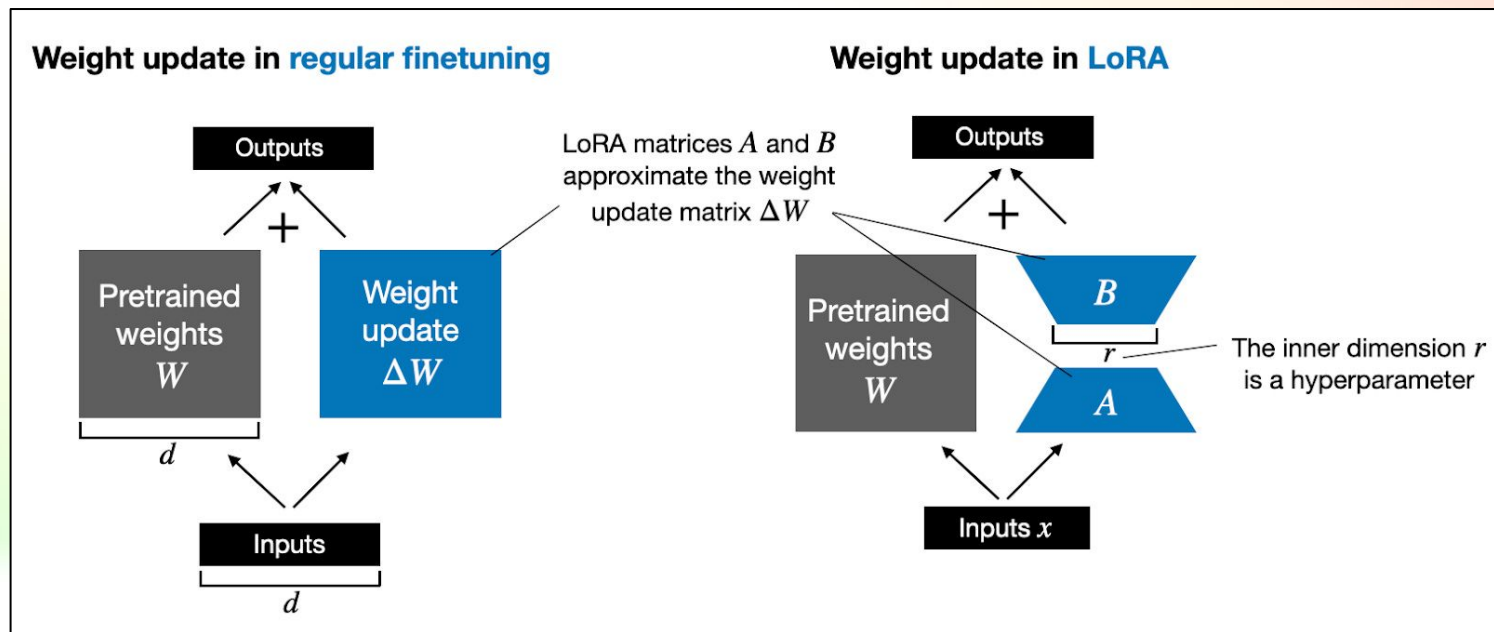
- oftentimes **boosts performance** on particular domain/task
- E.g. [data extraction](#)



The rise of (open) LLMs

How to fine-tune?

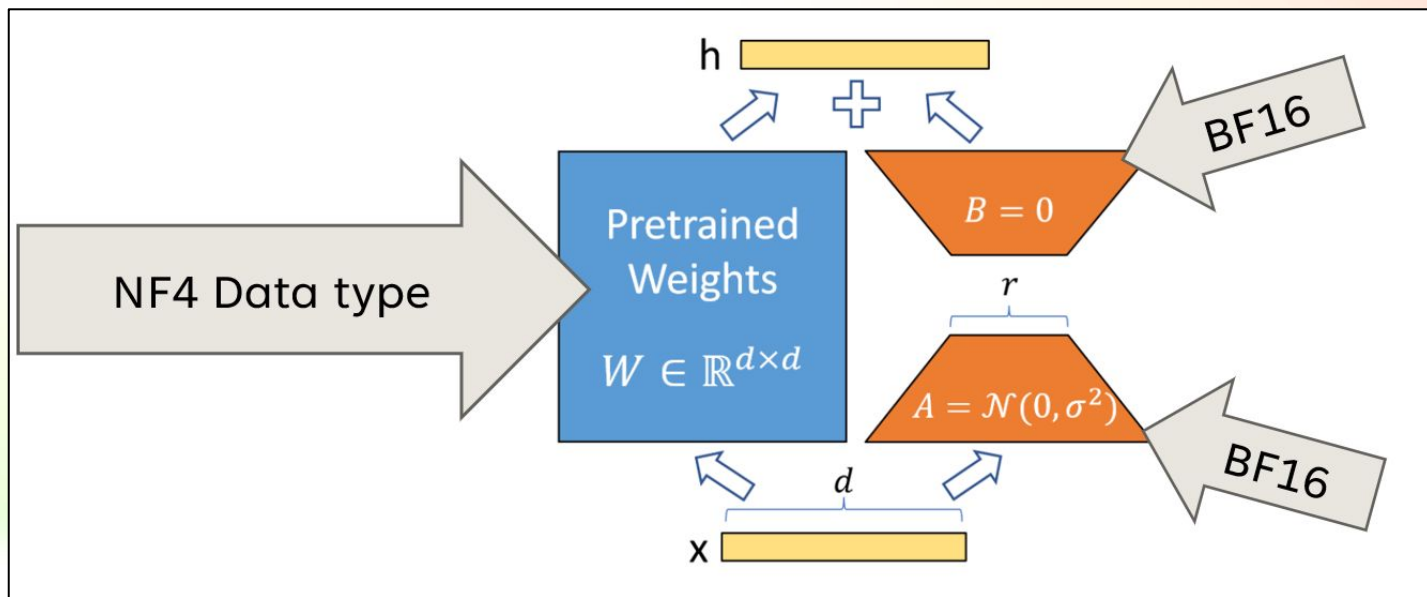
- LoRa



The rise of (open) LLMs

How to fine-tune?

- Q-LoRa



The rise of (open) LLMs

How to fine-tune?

- Tooling:
 - PEFT
 - TRL
- Highly recommended:
 - [Alignment Handbook](#)

Supervised Fine-tuning Trainer

All models SFT

Supervised fine-tuning (or SFT for short) is a crucial step in RLHF. In TRL we provide an easy-to-use API to create your SFT models and train them with few lines of code on your dataset.

Check out a complete flexible example at [examples/scripts/sft.py](#). Experimental support for Vision Language Models is also included in the example [examples/scripts/sft_vlm.py](#).

Quickstart

If you have a dataset hosted on the 🗄️ Hub, you can easily fine-tune your SFT model using [SFTTrainer](#) from TRL. Let us assume your dataset is `imdb`, the text you want to predict is inside the `text` field of the dataset, and you want to fine-tune the `facebook/opt-350m` model. The following code-snippet takes care of all the data pre-processing and training for you:

```
from datasets import load_dataset
from trl import SFTConfig, SFTTrainer

dataset = load_dataset("stanfordnlp/imdb", split="train")

training_args = SFTConfig(
    max_seq_length=512,
    output_dir="/tmp",
)
trainer = SFTTrainer(
    "facebook/opt-350m",
    train_dataset=dataset,
    args=training_args,
)
trainer.train()
```

The rise of (open) LLMs

How to deploy?

- Serverless vs. self-hosted



The rise of (open) LLMs

Serverless

- Everything is managed for you
- Charge per token
 - E.g. \$1 per 1 million tokens

API providers:

- OpenAI/Azure OpenAI
- Anthropic/AWS Bedrock
- Google/Vertex AI
- Together
- Fireworks
- Groq
- ...

```
import os
from openai import OpenAI

client =
OpenAI(api_key=os.environ.get("OPENAI_API_KEY"))

chat_completion = client.chat.completions.create(
    messages=[
        {
            "role": "user",
            "content": "How are you?",
        }
    ],
    model="gpt-4o",
)
```



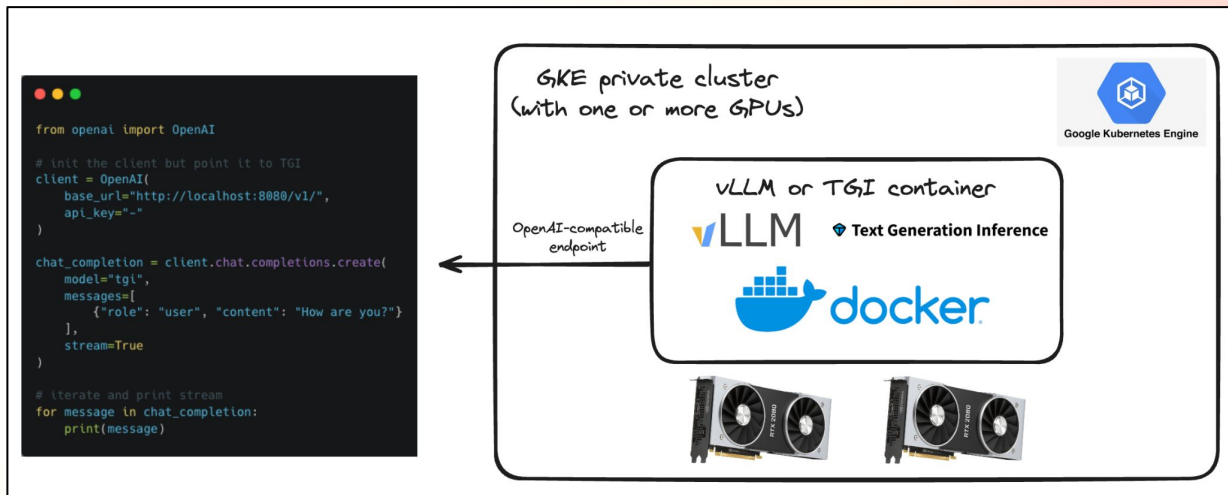
The rise of (open) LLMs

Self-hosted

- You own the model
- Charge per GPU hours
 - E.g. \$1 per GPU hour

Tooling:

- vLLM/TGI
- Docker
- Kubernetes/Cloud Run (GCP)
- 1 or more GPUs





The rise of (open) LLMs

Self-hosted

- You own the model
- Charge per GPU hours
 - E.g. \$1 per GPU hour

Tooling:

- vLLM/TGI
- Docker
- Kubernetes/Cloud Run (GCP)
- 1 or more GPUs

Google Kubernetes Engine (GKE) > Documentation > Guides Was this helpful?  

Serve Gemma open models using GPUs on GKE with vLLM

[Send feedback](#)

[AUTOPILOT](#) [STANDARD](#)

This tutorial shows you how to serve a [Gemma](#) large language model (LLM) using graphical processing units (GPUs) on Google Kubernetes Engine (GKE) with the [vLLM](#) serving framework.

In this tutorial, you download a [Gemma 2](#) (2B, 9B, and 27B parameter) instruction tuned model from Hugging Face. You then deploy the model on a GKE [Autopilot](#) or [Standard](#) cluster using a container that runs vLLM.

This guide is a good starting point if you need the granular control, scalability, resilience, portability, and cost-effectiveness of managed Kubernetes when deploying and serving your AI/ML workloads. If you need a unified managed AI platform to rapidly build and serve ML models cost effectively, we recommend that you try our [Vertex AI](#) deployment solution.

[Source](#)



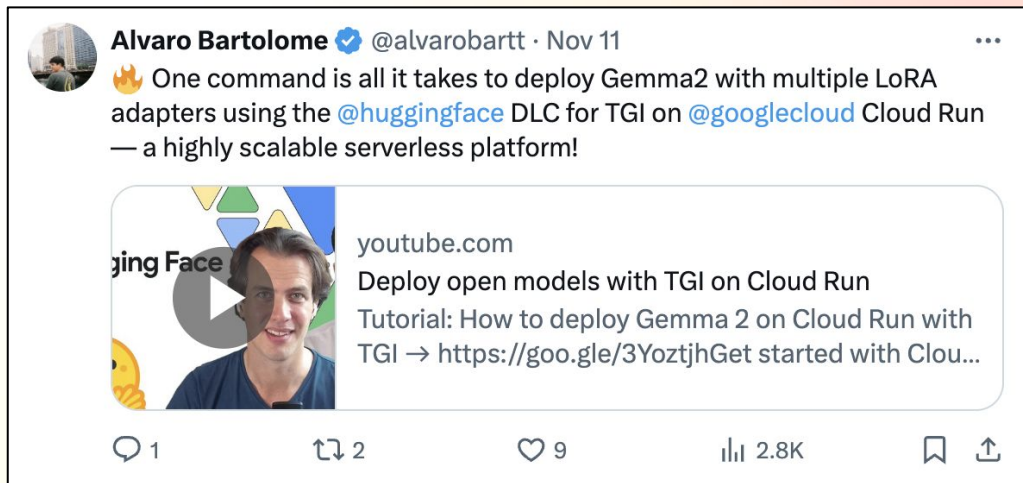
The rise of (open) LLMs

Self-hosted

- You own the model
- Charge per GPU hours
 - E.g. \$1 per GPU hour

Tooling:

- vLLM/TGI
- Docker
- Kubernetes/Cloud Run (GCP)
- 1 or more GPUs



The rise of (open) LLMs

Self-hosted

- You own the model
- Charge per GPU hours
 - E.g. \$1 per GPU hour

- Once deployed, can be called using **OpenAI API**

OpenAI Compatible Server

vLLM provides an HTTP server that implements OpenAI's [Completions](#) and [Chat](#) API.

You can start the server using Python, or using [Docker](#):

```
vllm serve NousResearch/Meta-Llama-3-8B-Instruct --dtype auto --api-key token-abc123
```

To call the server, you can use the official OpenAI Python client library, or any other HTTP client.

```
from openai import OpenAI
client = OpenAI(
    base_url="http://localhost:8000/v1",
    api_key="token-abc123",
)

completion = client.chat.completions.create(
    model="NousResearch/Meta-Llama-3-8B-Instruct",
    messages=[
        {"role": "user", "content": "Hello!"}
    ]
)

print(completion.choices[0].message)
```

Source: [vLLM](#)

The rise of (open) LLMs

Self-hosted

- You own the model
- Charge per GPU hours
 - E.g. \$1 per GPU hour

- Really cool use case:
 - **Multi-LoRa serving**

TGI Multi-LoRA: Deploy Once, Serve 30 models

Published July 18, 2024

[Update on GitHub](#)



[derek-thomas](#)
Derek Thomas



[dmaniloff](#)
Diego Maniloff



[dxbh](#)
David Holtz

Are you tired of the complexity and expense of managing multiple AI models? **What if you could deploy once and serve 30 models?** In today's ML world, organizations looking to leverage the value of their data will likely end up in a *fine-tuned world*, building a multitude of models, each one highly specialized for a specific task. But how can you keep up with the hassle and cost of deploying a model for each use case? The answer is Multi-LoRA serving.

[Blog post](#)



The rise of (open) LLMs

Self-hosted

- You own the model
- Charge per GPU hours
 - E.g. \$1 per GPU hour

- Really cool use case:
 - **Multi-LoRa serving**

Base model
(e.g. Llama-8B)

LoRa for SQL
generation

LoRa for translation

LoRa for
summarization

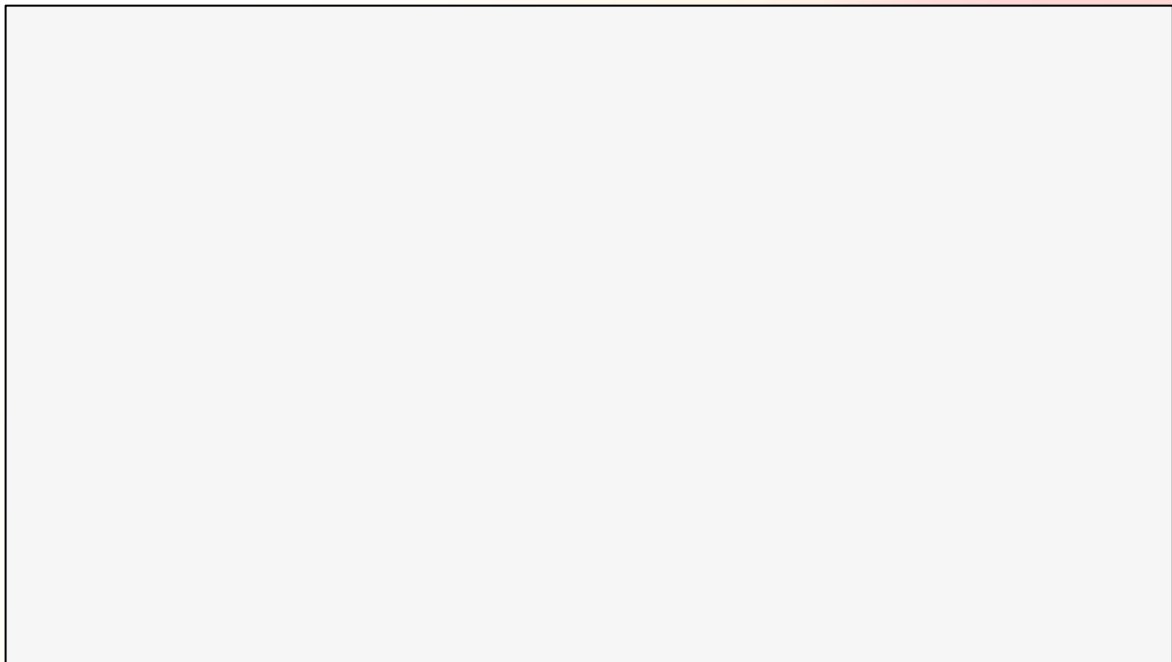


The rise of (open) LLMs

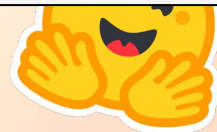
Self-hosted

- You own the model
- Charge per GPU hours
 - E.g. \$1 per GPU hour

- Really cool use case:
 - **Multi-LoRa serving**



Source: [Apple](#)



The rise of (open) LLMs

Self-hosted

- You own the model
- Charge per GPU hours
 - E.g. \$1 per GPU hour
- Really cool use case:
 - **Multi-LoRa serving**

Source: [Apple](#)



Current trends

LLMs:

- Train larger models
- Train small models
- Improve reasoning
- Train models to use your computer
- Extend with vision
- Extend with voice

Image/video generation:

- Impressive releases

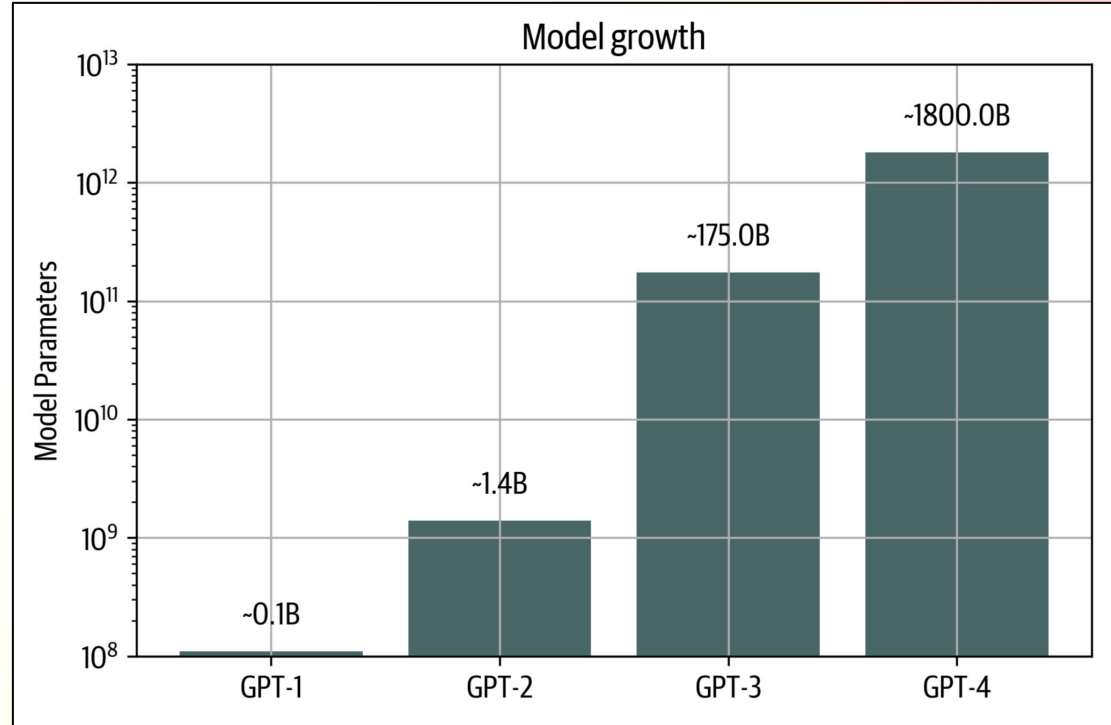
Robotics:

- Huge interest in humanoids



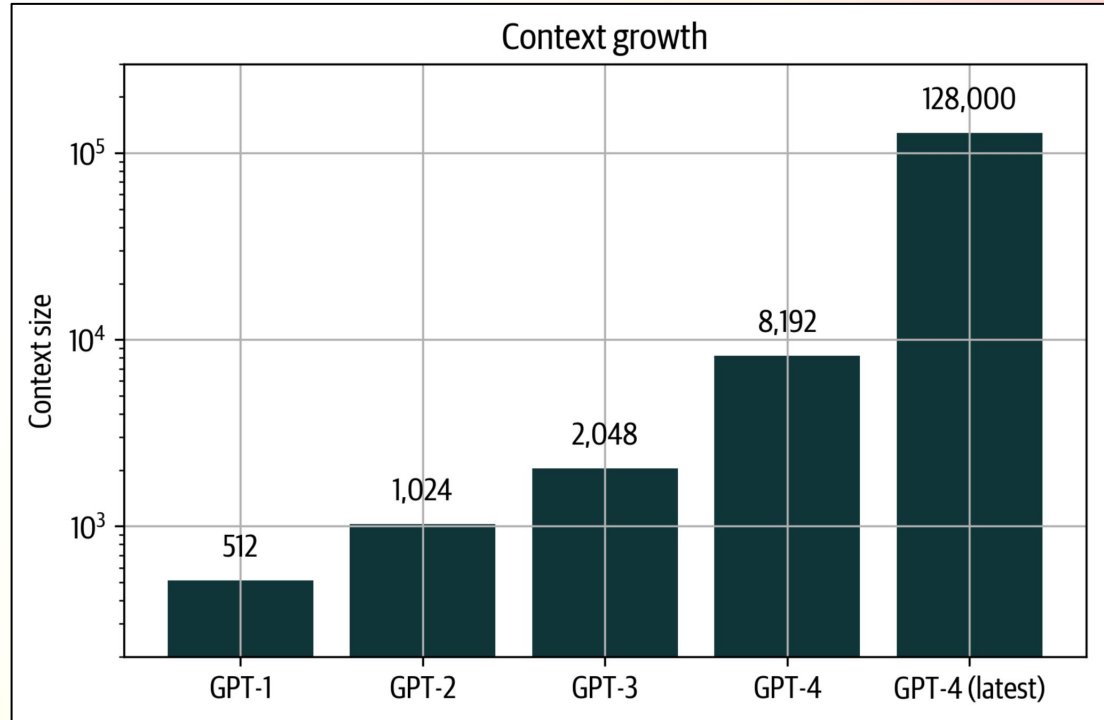
Current trends

- Train **larger** models
 - More parameters



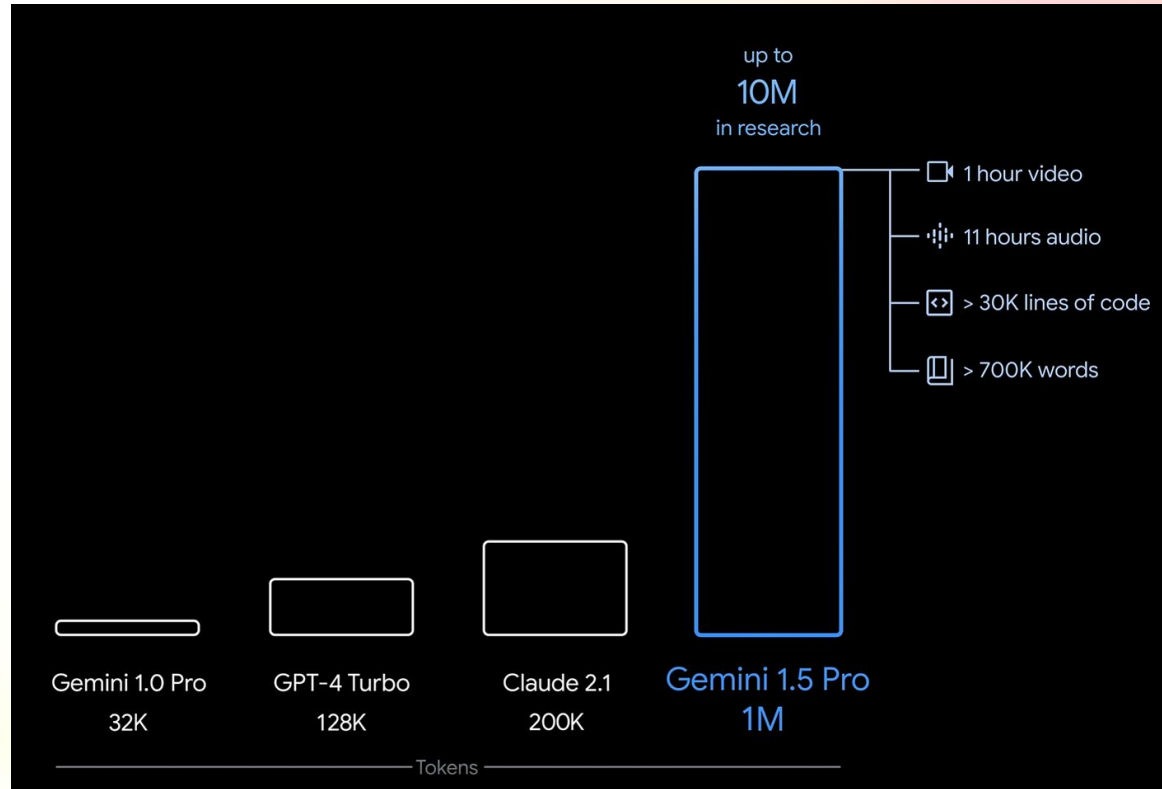
Current trends

- Train **larger** models
 - Longer context



Current trends

- Train **larger** models
 - Longer context



Current trends

- Train **larger** models
 - More compute

compute \approx **data** \times **model size**

| | Dataset (Billion Tokens) | Model size (Billion Parameter) | |
|---------------|-----------------------------|-----------------------------------|---------|
| GPT 1: | 1-2 | 0.11 | } 100x |
| GPT 2: | 10-20 | 1.4 | |
| GPT 3: | 300 | 175 | } 2000x |
| GPT 4: | 10'000 | 1'800 | |

Compute:
100x
2000x
300x

↙ GPT-4 cost: **~\$100M Dollars**



Current trends

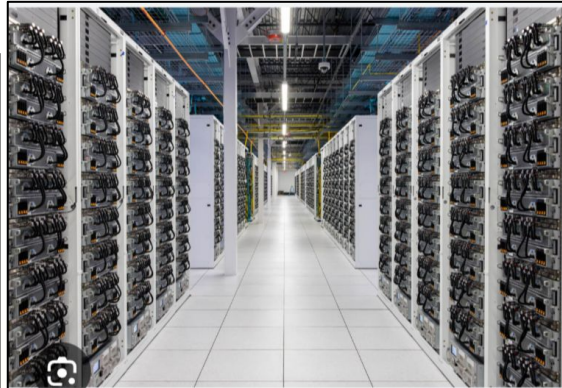
- Train **larger** models
 - More compute



Breaking news: In the past 24 hours, details were leaked about GPT-4. The information was analysis by [Dylan Patel](#) posted [here on SemiAnalysis](#), but put juicy details behind a paywall. [Yam Peleg](#) shared those details on Twitter, but then took down his tweet thread “due to a copyright claim.”

However, his information is [still available here](#), and we will summarize what we know about GPT-4 and what it means. Details are below, but at the top-line, we know:

- GPT-4 is a mixture-of-experts model, with 16 experts of 111B parameters each.
- It took about 2×10^{25} FLOPS to train, with 13 trillion token (passes).
- **Estimated pre-training hardware utilization cost of \$63 million, using 25,000 A100s almost 100 days to do the training.**
- The training and architecture was to optimize it for inference, and inference costs were about 3 times that of GPT-3 / DaVinci.



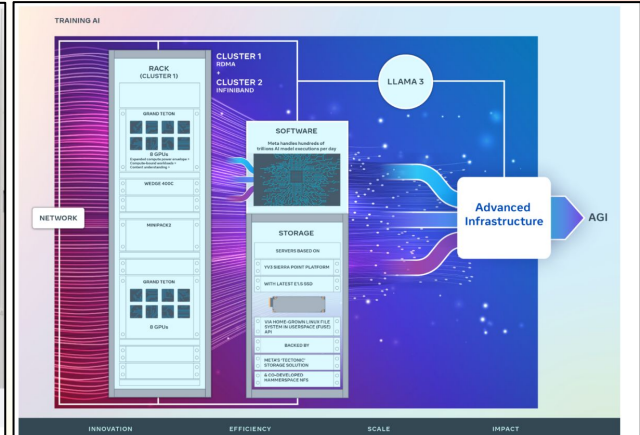
Inside the 100K GPU xAI Colossus Cluster that...

Visit >

Creator: patrick kennedy pa...

Want to know where this information comes from? [Learn more](#)

Images may be subject to copyright. [Learn More](#)



Under the hood

Our newer AI clusters build upon the successes and lessons learned from RSC. We focused on building end-to-end AI systems with a major emphasis on researcher and developer experience and productivity. The efficiency of the high-performance network fabrics within these clusters, some of the key storage decisions, combined with the **24,576 NVIDIA Tensor Core H100 GPUs in each**, allow both cluster versions to support models larger and more complex than that could be supported in the RSC and pave the way for advancements in GenAI product development and AI research.

Current trends

- Train **small** models

SmolLM - blazingly fast and remarkably powerful

Published July 16, 2024

[Update on GitHub](#)



[@loubnabn1](#)
Loubna Ben Allal



[@anton-1](#)
Anton Lozhkov



[@eliebak](#)
Elie Bakouch

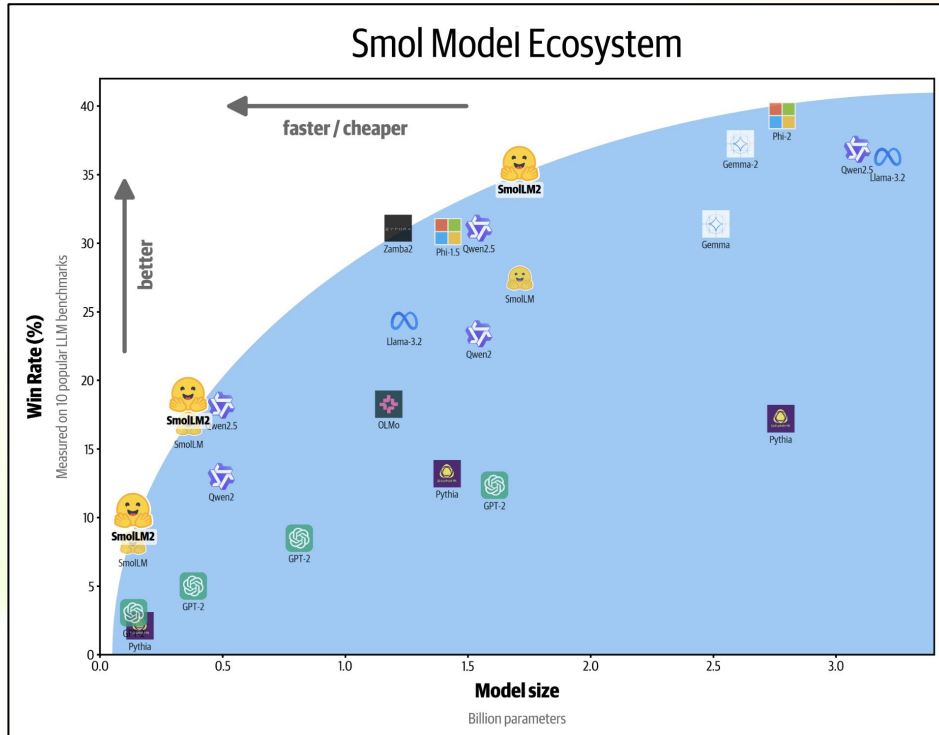
TL;DR

This blog post introduces [SmolLM](#), a family of state-of-the-art small models with 135M, 360M, and 1.7B parameters, trained on a new high-quality dataset. It covers data curation, model evaluation, and usage.



Current trends

- Train **small** models



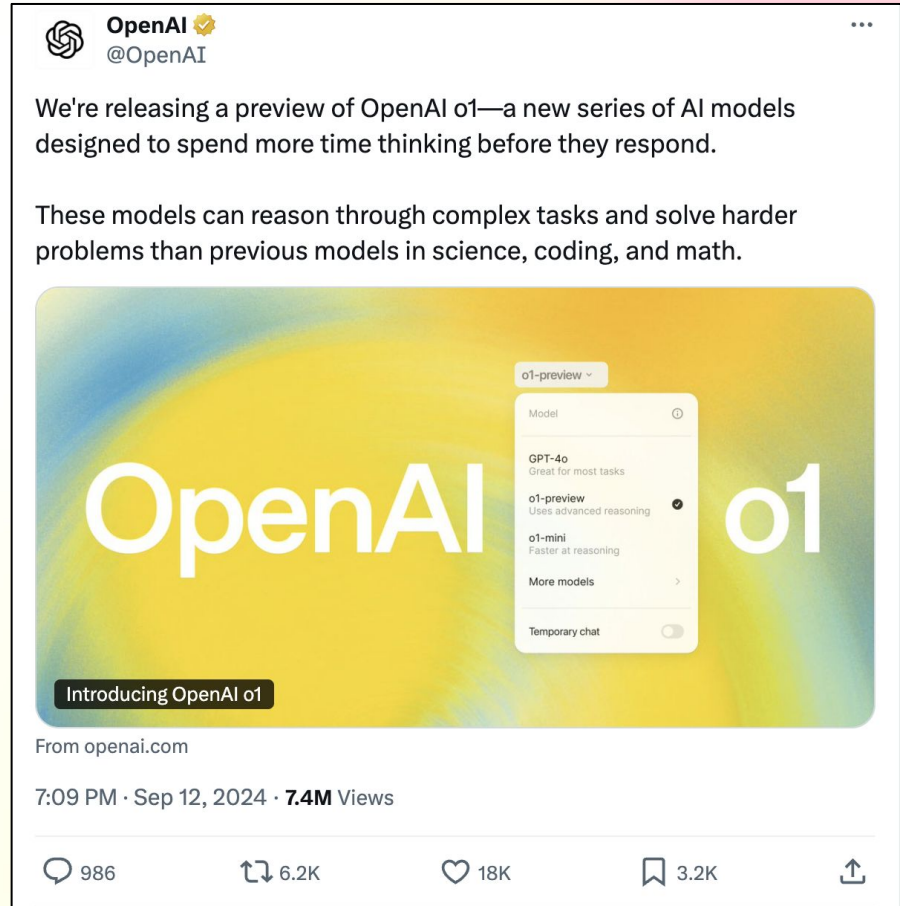
Current trends

- Train **small** models
- Demo: <https://x.com/awnihannun/status/1852400317717197254>



Current trends

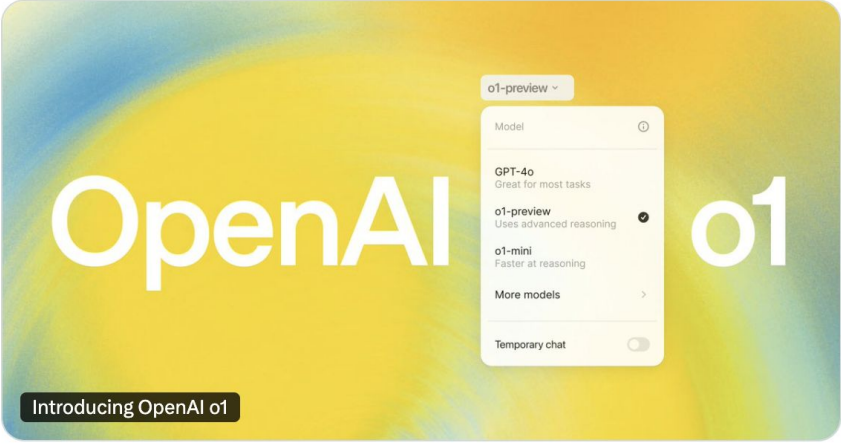
- Improve **reasoning**
- Go beyond text generation:
 - Problem solving
 - Sophisticated reasoning
 - Agents



OpenAI ✓
@OpenAI

We're releasing a preview of OpenAI o1—a new series of AI models designed to spend more time thinking before they respond.

These models can reason through complex tasks and solve harder problems than previous models in science, coding, and math.



The image shows a screenshot of the OpenAI chat interface. The background is a yellow and blue gradient with the text 'OpenAI o1' in large white letters. A dropdown menu is open, showing the following options: 'GPT-4o' (Great for most tasks), 'o1-preview' (Uses advanced reasoning, selected), 'o1-mini' (Faster at reasoning), 'More models' (with a right arrow), and 'Temporary chat' (with a toggle switch).

Introducing OpenAI o1

From openai.com


7:09 PM · Sep 12, 2024 · **7.4M** Views


986 6.2K 18K 3.2K







Current trends


- Improve **reasoning**
- Go beyond text generation:
 - Problem solving
 - Sophisticated reasoning
 - Agents

DeepSeek 
@deepseek_ai

 DeepSeek-R1-Lite-Preview is now live: unleashing supercharged reasoning power!

-  o1-preview-level performance on AIME & MATH benchmarks.
-  Transparent thought process in real-time.
-  Open-source models & API coming soon!

 Try it now at chat.deepseek.com
[#DeepSeek](#)

 **Hi, I'm DeepSeek.**

How can I help you today?

Message DeepSeek

Deep Think (50 messages left today) **NEW**

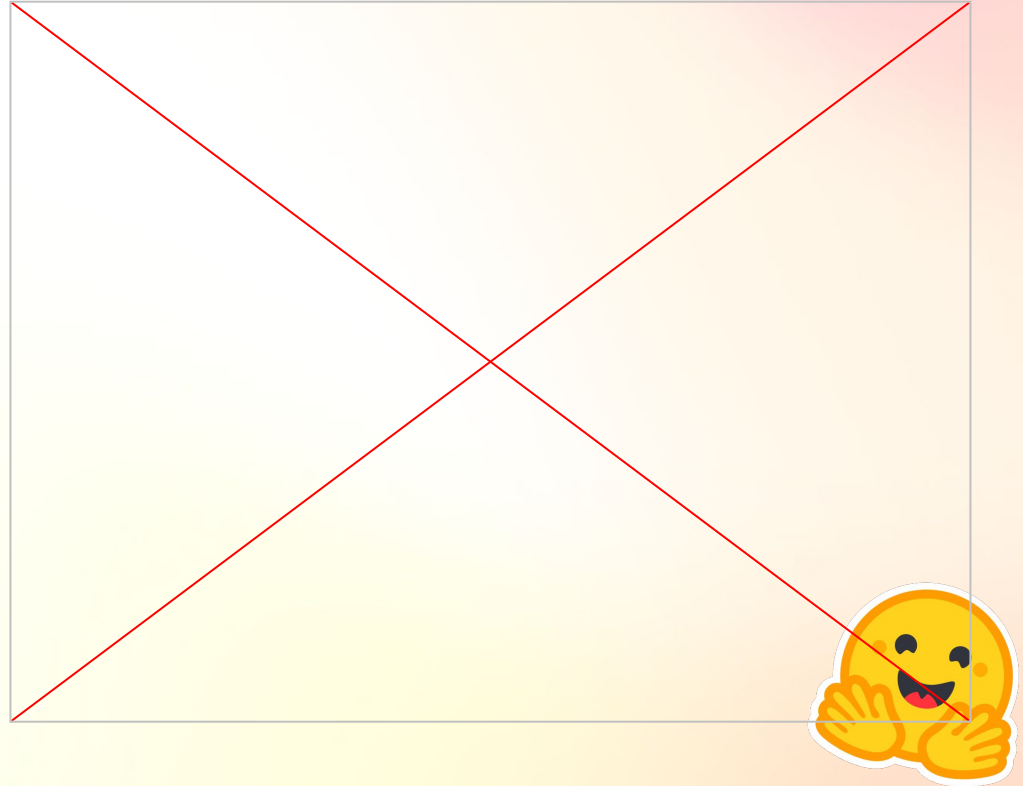
12:40 PM · Nov 20, 2024 · **404.4K** Views

130 597 1.8K 686



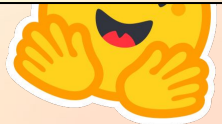
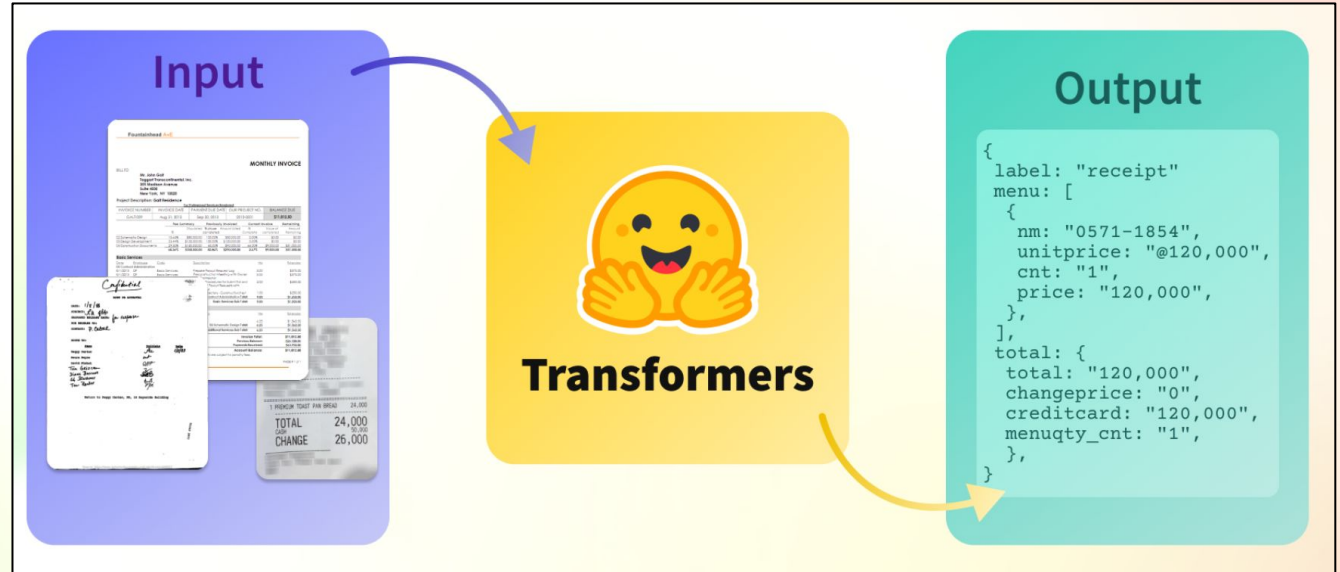
Current trends

- Train models to **use your computer**
 - Handle task autonomously
- Like RPA, but powered by a Transformer



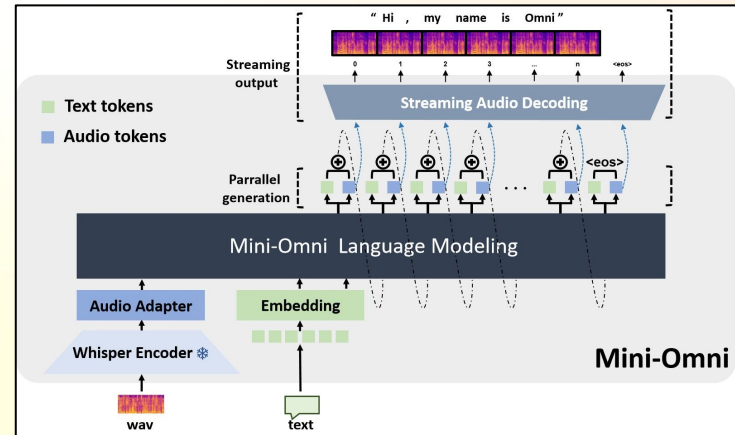
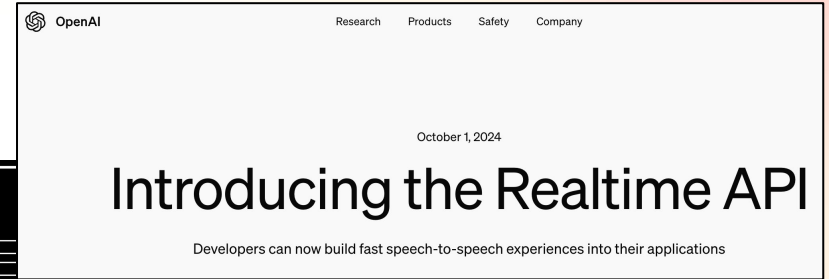
Current trends

- Extend with **vision**
 - Document AI



Current trends

- Extend with **voice**
 - Speech-to-speech use cases
 - “Talk with an AI”



Current trends

- Image/video generation:
- Flux: open model
 - Schnell
 - Dev
 - Pro



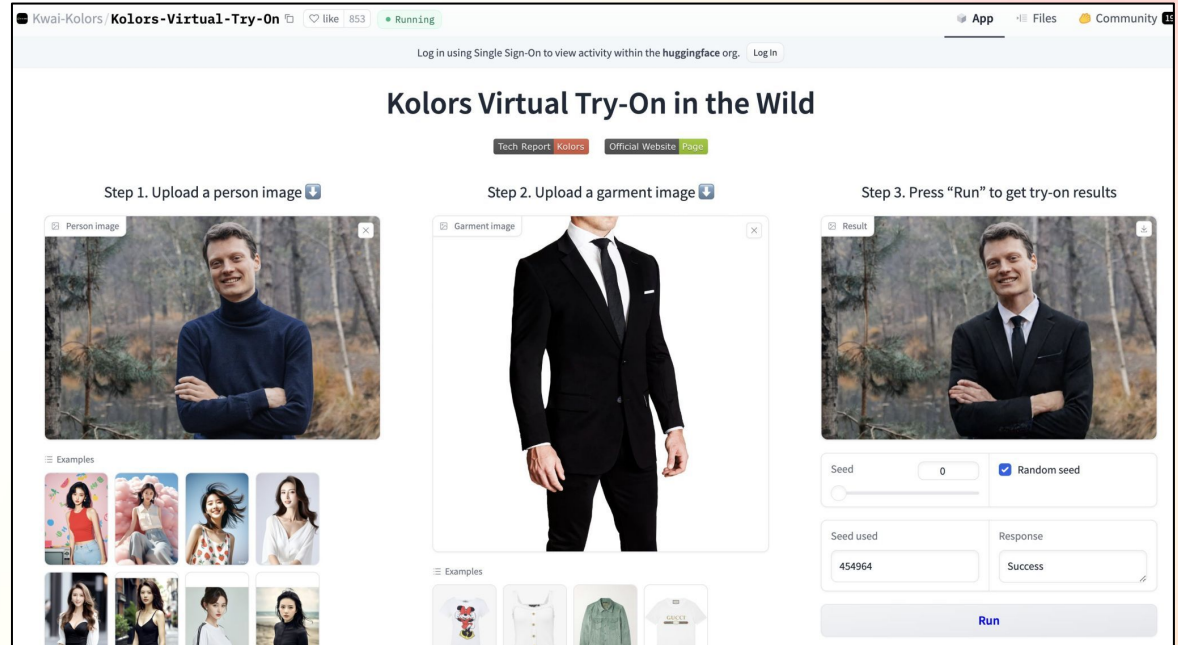
Current trends

- Image/video generation:
- Flux
 - Easy to fine-tune (LoRa)



Current trends

- Image/video generation:
 - Easy to fine-tune (LoRa)
- Flux
- Virtual try-on use cases



Current trends

- Image/video generation:
- Flux
 - Easy to fine-tune (LoRa)
- Virtual try-on use cases

 **Mickey Friedman** ✓
@mickeyxfriedman

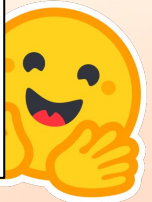
if you want to build midjourney for ecommerce, consider joining [@flairAI_](#)

team of 6, but since launching we've grown to over 1m users and served over 10,000 monthly paying customers

hiring 10x typescript devs (80% frontend, 20% backend) - dms are open if this is you



12:11 AM · Sep 7, 2024 · **32.3K** Views



Current trends

- Image/video generation:
- Flux
 - Easy to fine-tune (LoRa)
- Product marketing



Current trends

- Image/video generation:
- Meta's MovieGen, Kling.ai, Runway,...

“A woman DJ spins records on a rooftop in LA. She is wearing a pink jacket and giant headphones. There is a cheetah next to the woman. The background is a cityscape.”



<https://ai.meta.com/research/movie-gen/>



Current trends

- Image/video generation:
- Meta's MovieGen, Kling.ai, Runway,...

Flair.ai combined with Kling.ai

Kling AI Video



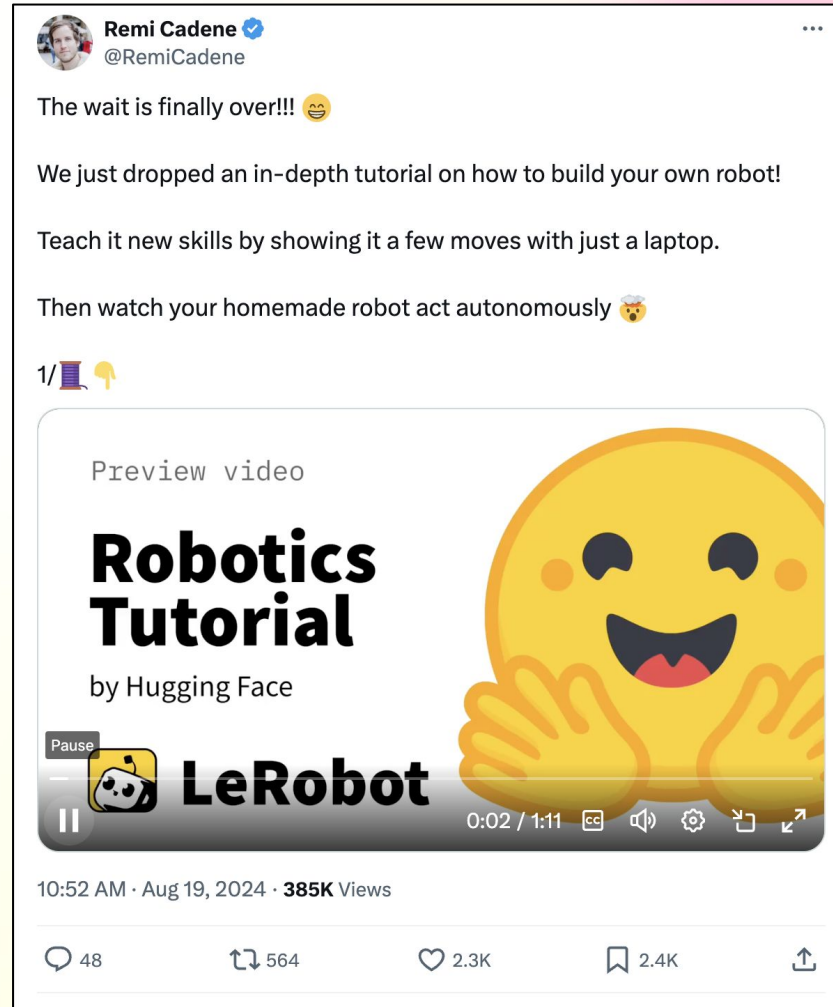
Current trends

- Robotics
- Huge interest in humanoids



Current trends

- Robotics
- LeRobot: democratizing robotics



Remi Cadene ✓
@RemiCadene

The wait is finally over!!! 🥳

We just dropped an in-depth tutorial on how to build your own robot!

Teach it new skills by showing it a few moves with just a laptop.

Then watch your homemade robot act autonomously 🤖

1/📖👉

Preview video

Robotics Tutorial

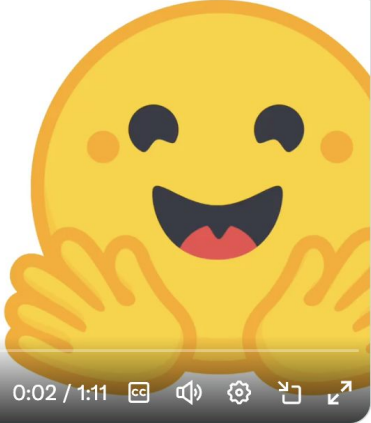
by Hugging Face

LeRobot

0:02 / 1:11

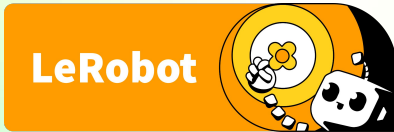
10:52 AM · Aug 19, 2024 · 385K Views

48 564 2.3K 2.4K




Current trends

- Robotics
- LeRobot: democratizing robotics



Business cases

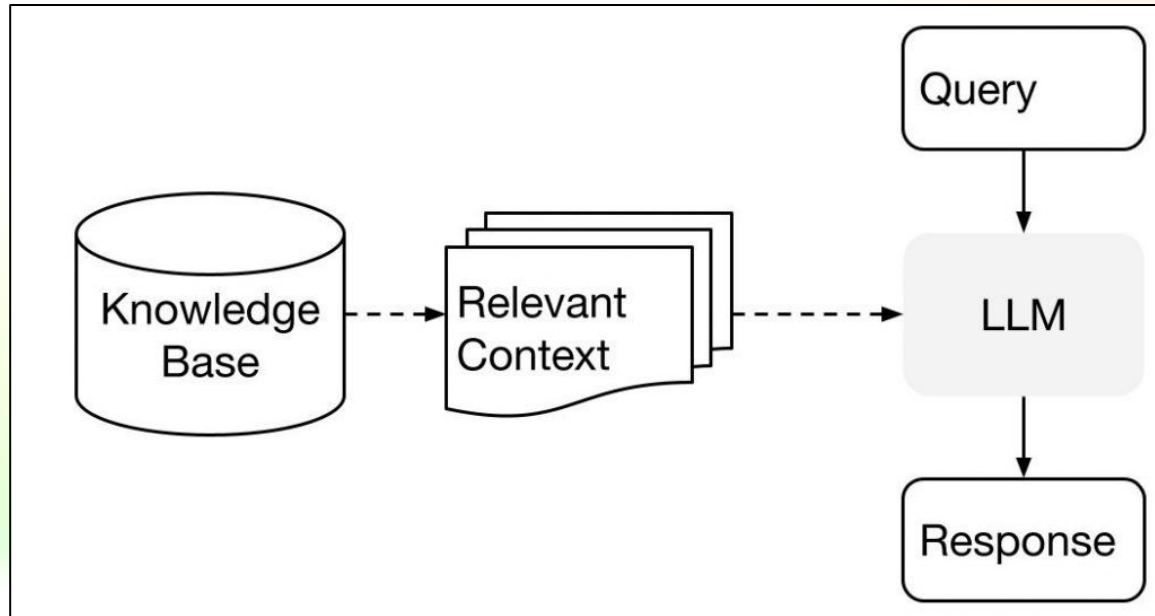
Most popular ones at 

- RAG chatbots
- Structured data extraction
- Voice agents



Business cases


- RAG chatbots
 - Allows to **“chat with your data”**




Business cases

- RAG chatbots
 - Allows to **“chat with your data”**
 - Useful for
 - Internal employees
 - Customer support
- Benefits of RAG:
 - Grounded answers
 - Reduced hallucinations


LLM Chatbot RAG Assistant

 How do I drain the filter?

 Sure, here's a detailed answer to the question:

To drain the filter, follow these steps:

1. Turn off the washer and unplug the power cord.
2. Gently press the top area of the filter cover to open it.
3. Drain the remaining water inside the drum using the emergency drain tube. Refer to the emergency drain section on page 42.
4. Turn the drain filter knob counterclockwise and drain the remaining water.
5. Clean the drain filter using a soft brush while the filter is removed. Ensure that the impeller inside the drain pump is free of debris.
6. Reinsert the drain filter and turn the knob clockwise to secure it.
7. Close the filter cover.

 How many times should I clean the drain filter?

 According to the context, maintenance recommends cleaning the drain filter 5 or 6 times a year.

Ask me anything! 



Business cases

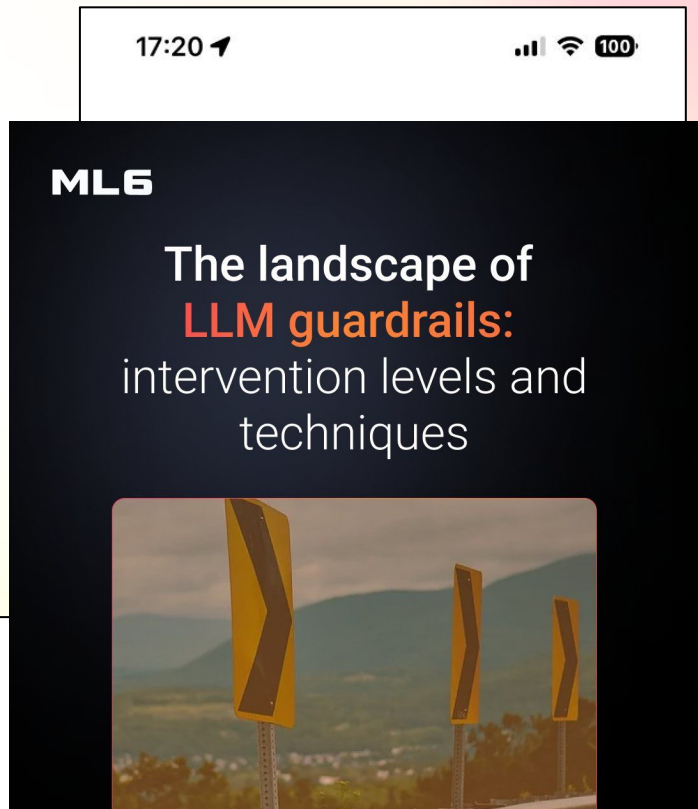
- RAG chatbots
 - Allows to “chat with your data”
 - Useful for
 - Internal employees
 - Customer support
- **Guardrails:** important!
 - Avoid situations like this:

 The Guardian

DPD AI chatbot swears, calls itself 'useless' and criticises delivery firm

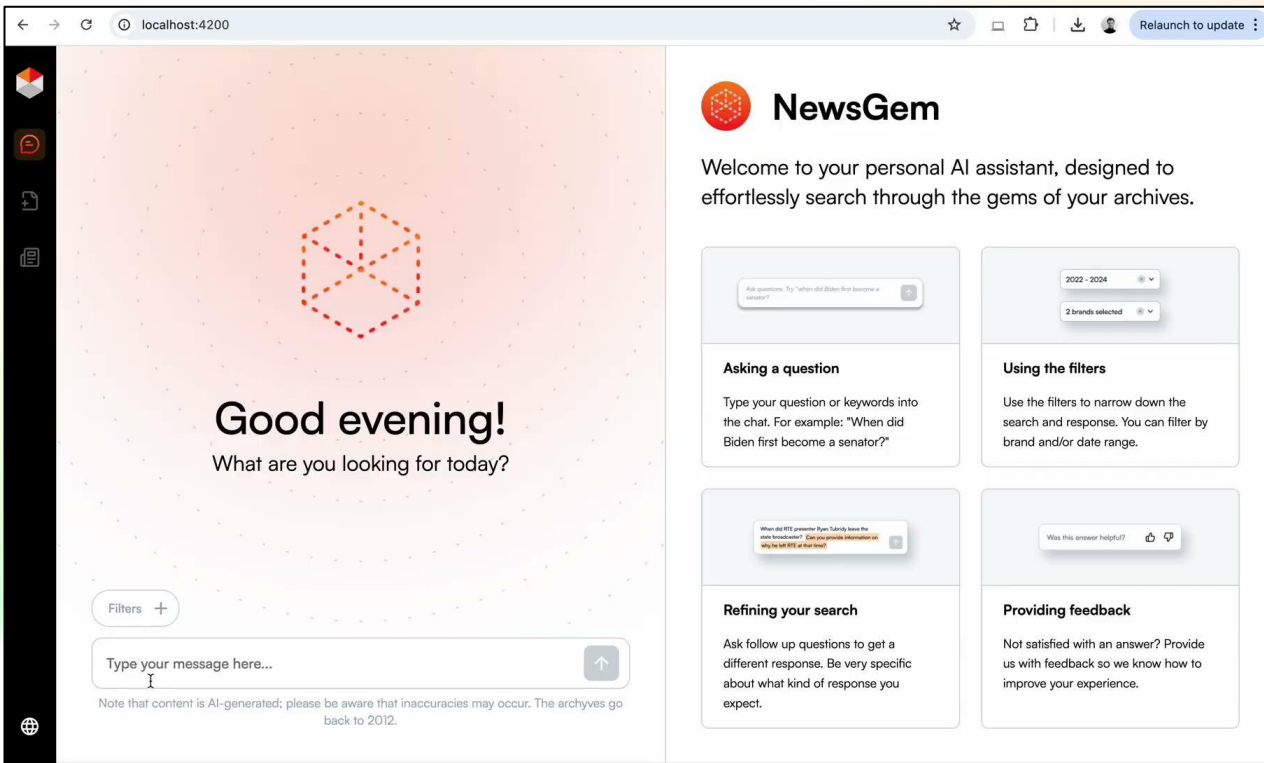
DPD AI chatbot swears, calls itself 'useless' and criticises delivery firm ... The delivery firm DPD has disabled part of its artificial...

20 Jan 2024



Business cases

- RAG chatbot developed at 



The screenshot shows a web browser window at localhost:4200 displaying the NewsGem AI assistant interface. The interface is split into two main sections: a chat area on the left and a help/tutorial area on the right.

Chat Area (Left):

- Header: "Good evening!" followed by "What are you looking for today?"
- Input field: "Type your message here..." with a "Filters +" button above it.
- Footer: "Note that content is AI-generated; please be aware that inaccuracies may occur. The archives go back to 2012."

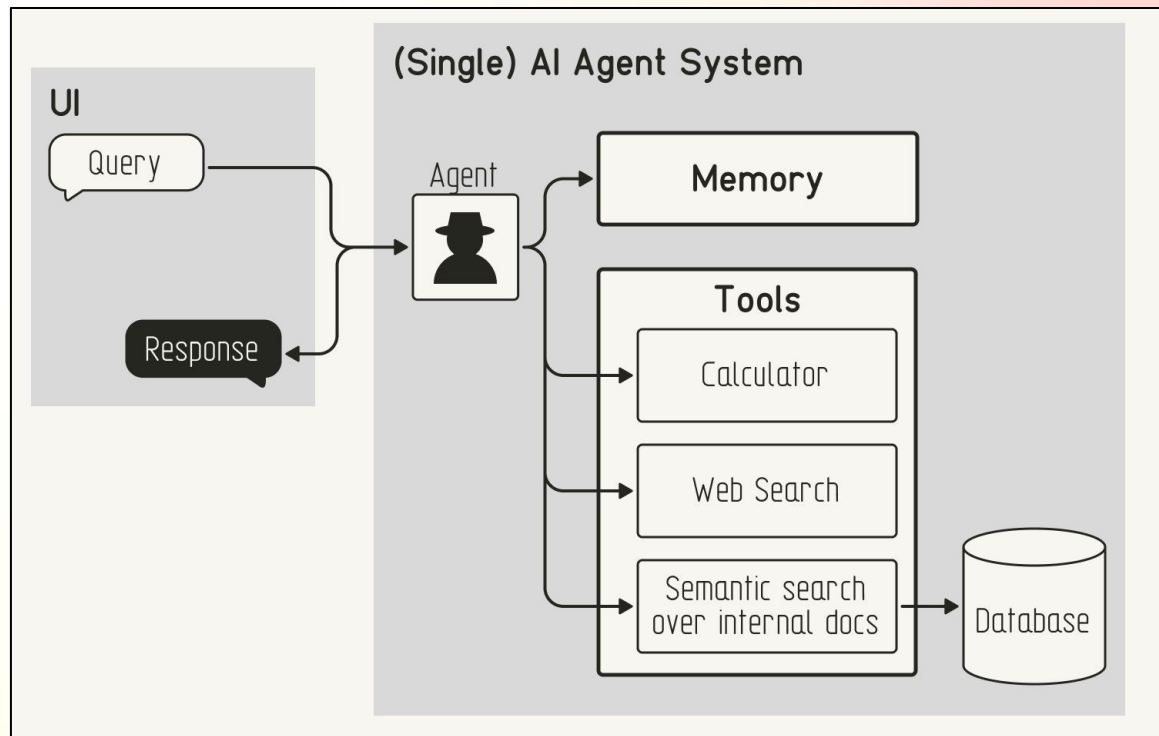
Help Area (Right):

- Logo: "NewsGem" with a red hexagonal icon.
- Welcome message: "Welcome to your personal AI assistant, designed to effortlessly search through the gems of your archives."
- Four instructional cards:
 - Asking a question:** "Ask questions. Try 'when did Biden first become a senator?'"
 - Using the filters:** "Use the filters to narrow down the search and response. You can filter by brand and/or date range." (Includes filters for "2022 - 2024" and "2 brands selected")
 - Refining your search:** "Ask follow up questions to get a different response. Be very specific about what kind of response you expect." (Includes a sample question: "When did FIE presenter Ryan Tubridy leave the show broadcast? Can you provide alternative dates? Why the SAT FIE at that time?")
 - Providing feedback:** "Not satisfied with an answer? Provide us with feedback so we know how to improve your experience." (Includes a "Was this answer helpful?" button)



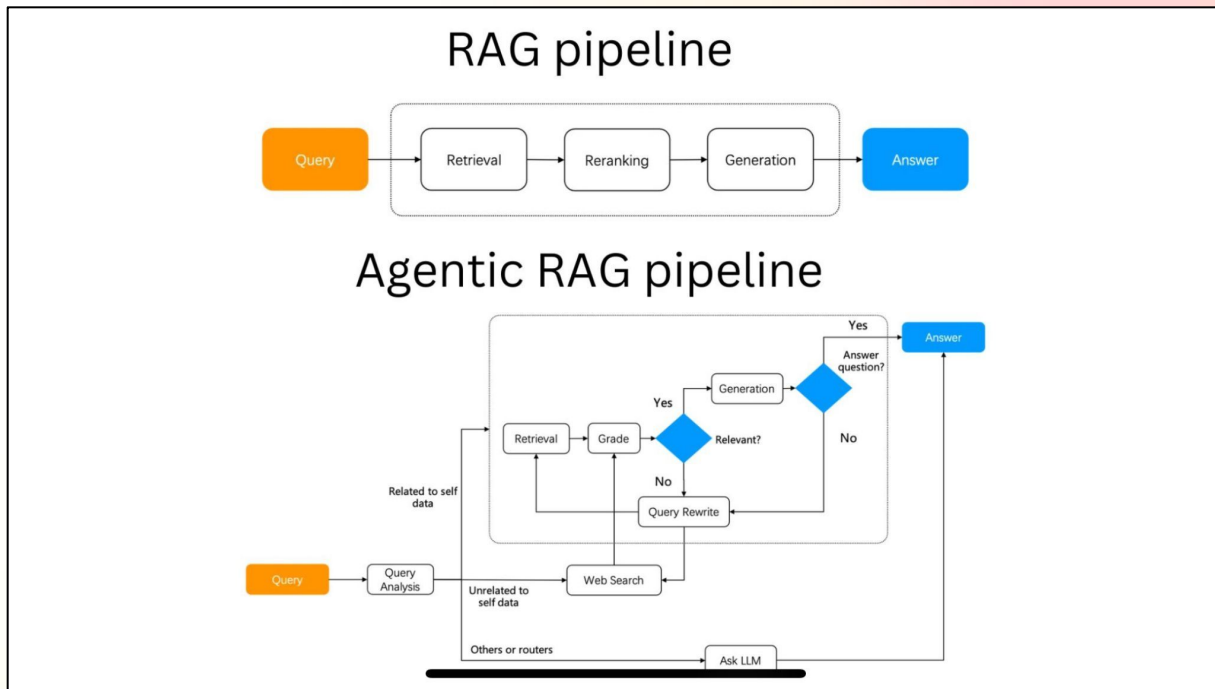
Business cases

- Agentic RAG
 - Generalization of basic RAG
 - LLM has access to a set of tools
 - LLM decides itself which tool to call and when



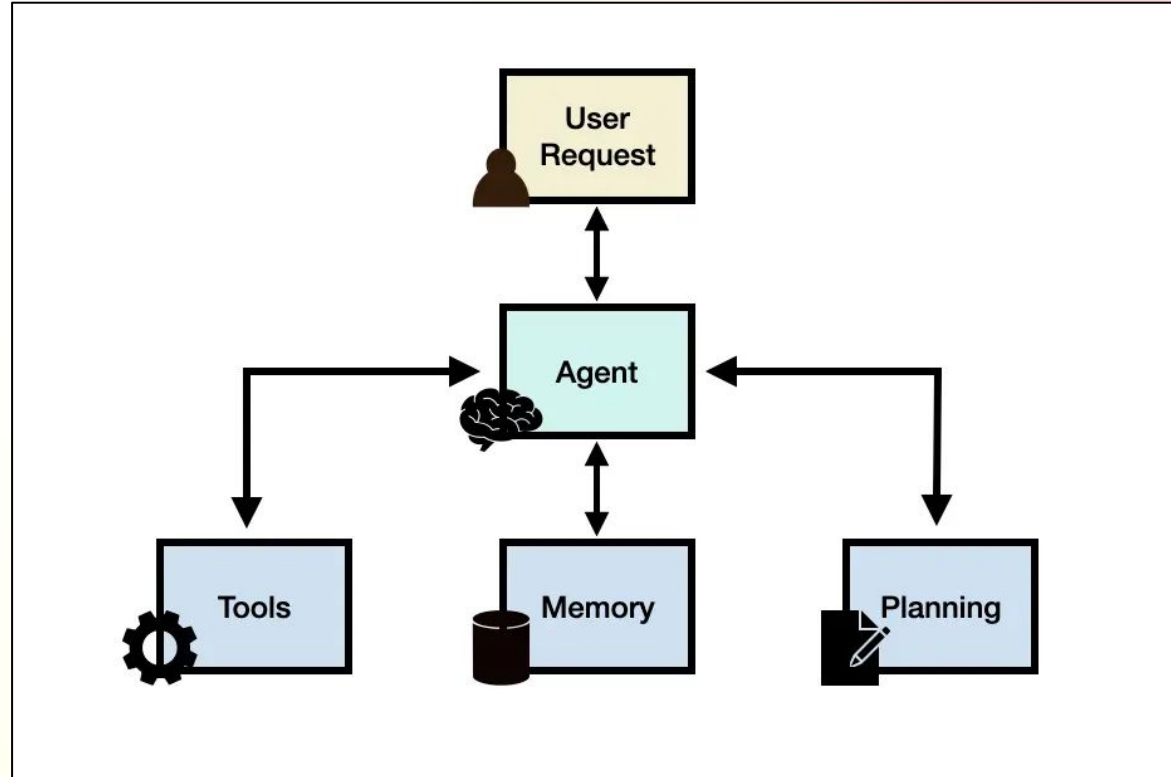
Business cases

- Agentic RAG
 - Generalization of basic RAG
 - LLM has access to a set of tools
 - LLM decides itself which tool to call and when



Business cases

- What is an **agent**?
 - LLM which has access to a set of **tools**
 - E.g. weather API
 - E.g. Google API
 - E.g. calculator
 - Can repeatedly call tools to achieve a certain goal
 - Often augmented with **memory** and **planning** (sketchpad)



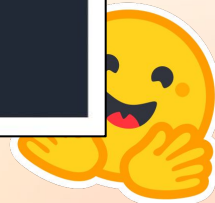
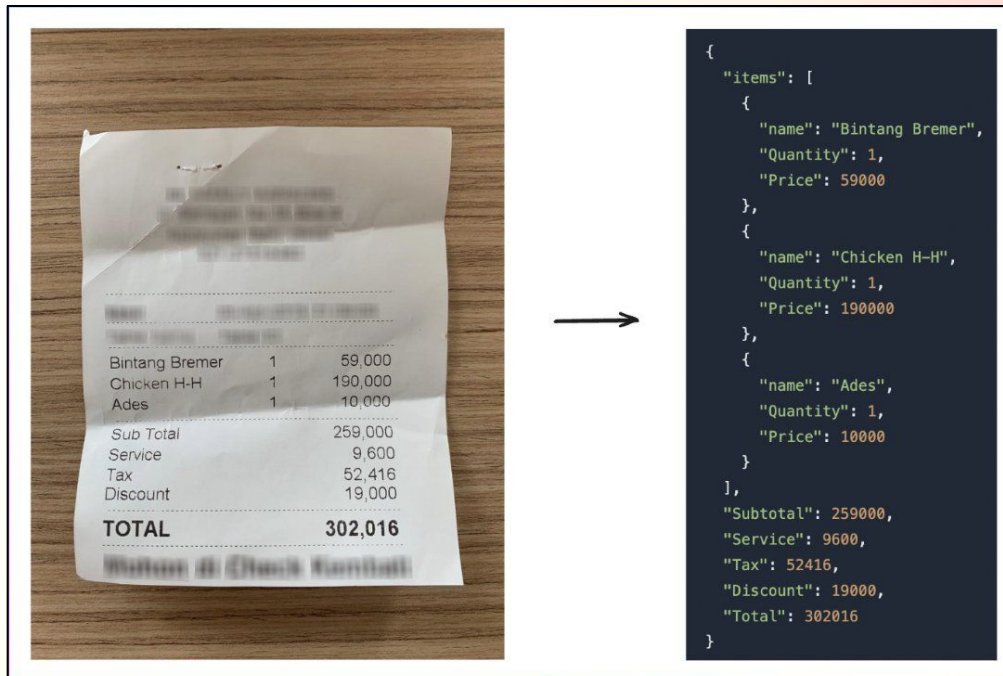
Business cases

- Multimodal RAG
 - Generalization of RAG to images + text



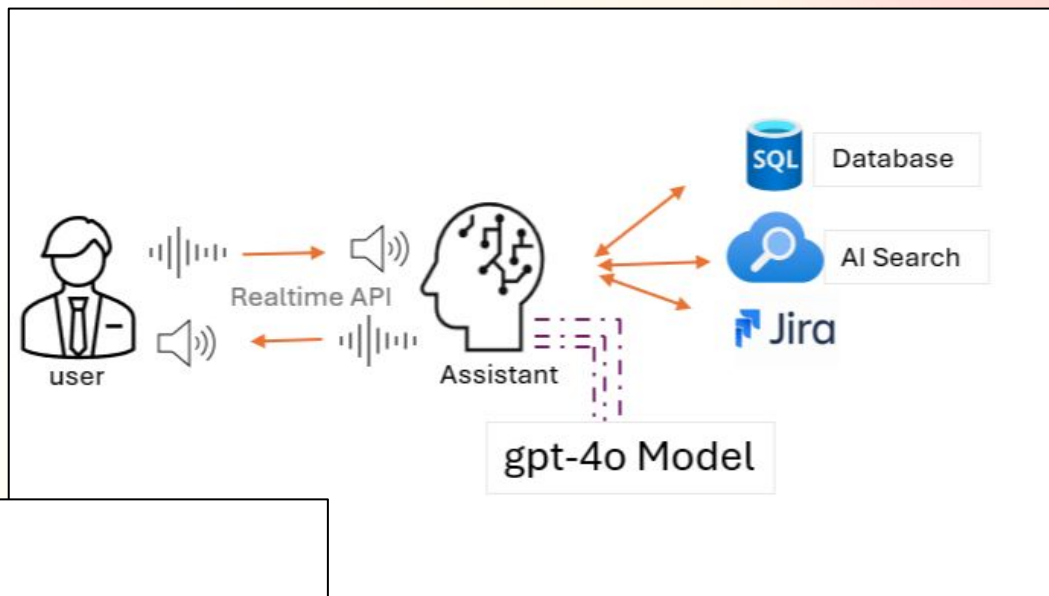
Business cases

- Structured data extraction
 - E.g. receipt → JSON
 - E.g. PDF → JSON
- Document AI
- Allows to parse key fields from documents



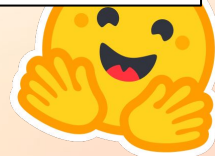
Business cases

- Voice agents
 - Audio in, audio out
- Use cases:
 - HR screening
 - Receptionist
 - Arrange appointments for business owners



Realtime API Beta

Build low-latency, multi-modal experiences with Realtime API.

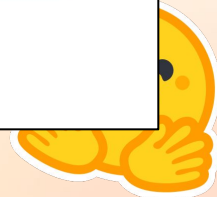


Business cases

- Voice agents
 - Audio in, audio out
- Use cases:
 - HR screening
 - Receptionist
 - Arrange appointments for business owners

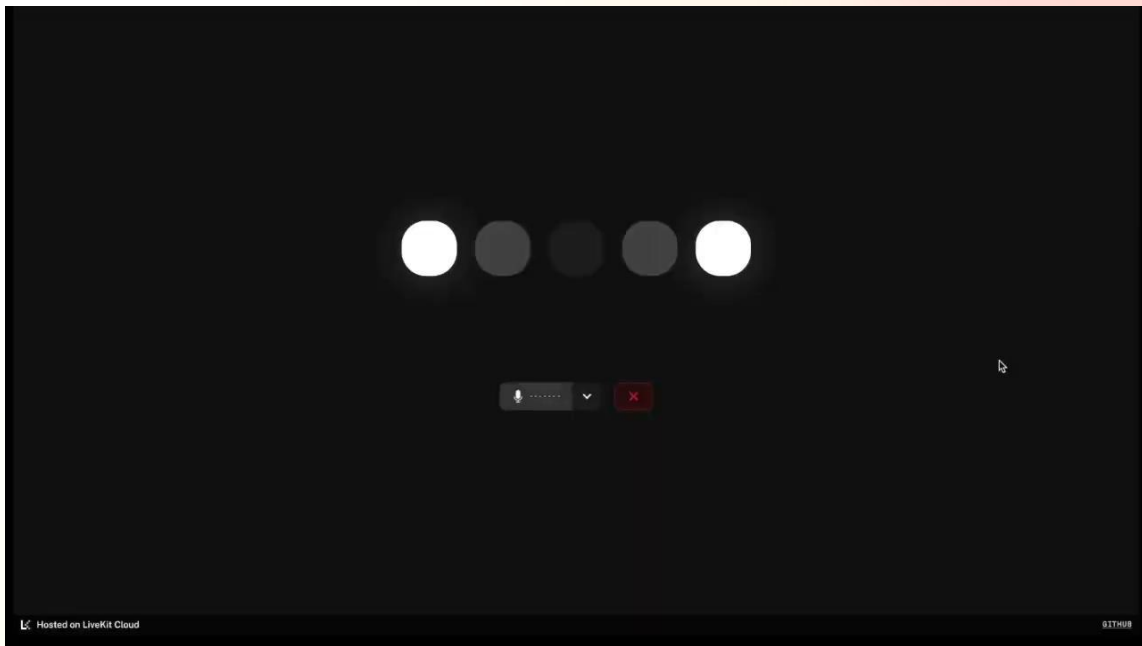
Google Duplex: An AI System for Accomplishing Real-World Tasks Over the Phone

May 8, 2018 · Posted by Yaniv Leviathan, Principal Engineer and Yossi Matias, Vice President, Engineering, Google



Business cases

- Voice agents
 - Audio in, audio out
- Use cases:
 - HR screening
 - Receptionist
 - Arrange appointments for business owners



Thanks for your attention!

PS: connect with me!
@NielsRogge

