

The landscape of pen LLMs

Leandro von Werra

Chief Loss Officer at Hugging Face

What does it take to **train an LLM**?



Model
(Transformer)

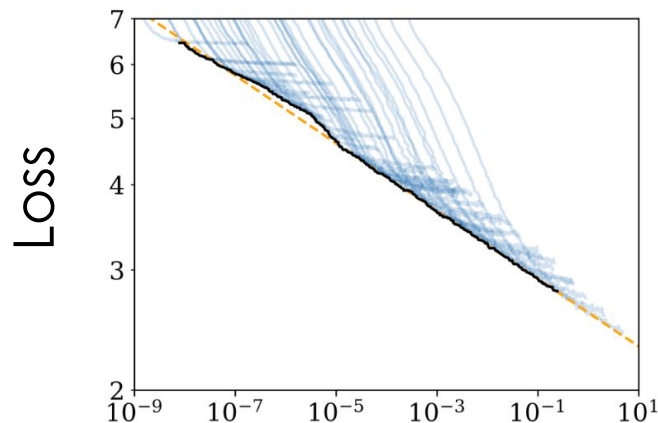


Data
(Web data etc.)

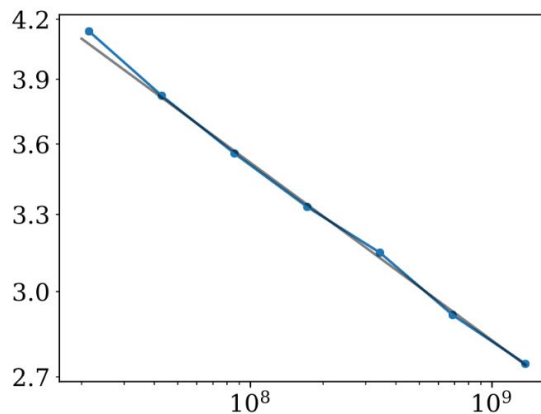


Compute
(aka GPUs, TPUs etc.)

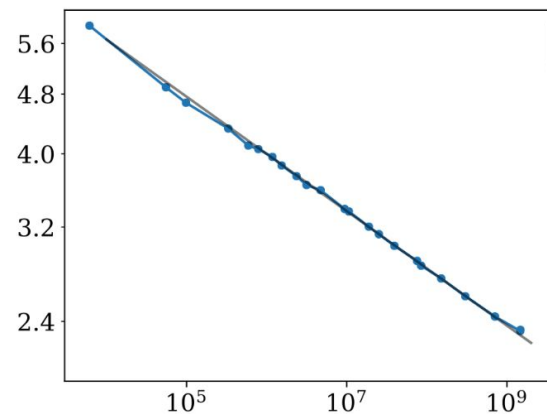
Early 2020: Scaling laws



Compute



Data



Model size

From GPT 1 → 4

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

↪ **GPT-4 cost: ~\$100M**

The **cost** in perspective



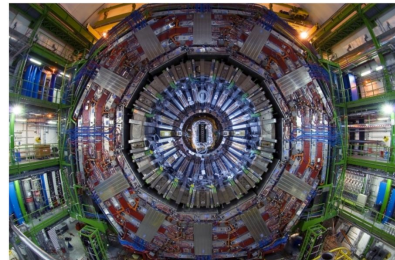
GPT-4:
~\$100M



787 airplane:
~\$100-200M



GPS satellite:
~\$200-500M



LHC:
~\$7'500M

What about **open** models?


 Data Center Dynamics

Meta to operate "600000 H100 GPU equivalents of compute" by year-end

Meta expects to field a fleet of 600,000 GPUs by the end of 2024. CEO Mark Zuckerberg told The Verge that the number includes some 340,000...

18 Jan 2024

****GPT-4 used 25'000 A100s for 3-4 months***

 Synced

DeepMind's Gemma: Advancing AI Safety and Performance with Open Models

Large Language Models (LLMs) have proven their mettle across a spectrum of real-world applications, ranging from language modeling to visual...



Mistral AI, a Paris-based OpenAI rival, closed its \$415 million funding round

Romain Dillet @romaindillet / 12:47 PM GMT+1 • December 11, 2023

 Comment



Open compute



LUMI (Finland): **11'912 GPUs** (AMD MI250x)



JUWELS (Germany): **3'774 GPUs** (NVIDIA A100)

coming soon

JUPITER (Germany): **24'000 GPUs** (NVIDIA GH200)



Leonardo (Italy): **13'824 GPUs** (NVIDIA A100)

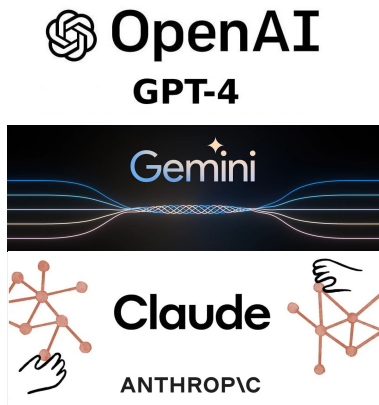


Alps (Switzerland): **10'000 GPUs** (NVIDIA GH100/200)

closed model APIs

model weights not available

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



open model weights

no access to training data or code

- who's data is in dataset?
- can't remove data on request
- can't inspect data for biases
- benchmark contamination
- limits scientific reproducibility



fully open model

full access to model/code/data

- competitive edge
- liability issues
- maintenance



Closed or **open** models?

“Software is eating the world.”

-Marc Andreessen, 2011

BigCode: open-scientific collaboration

We are building LLMs for code in a collaborative way:

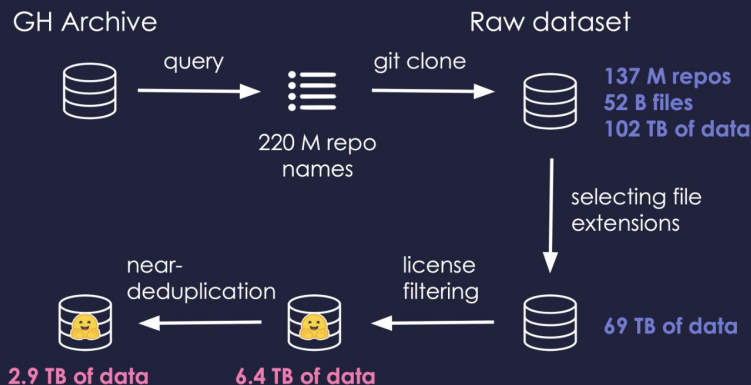
- Full data transparency
- Open source processing and training code
- Model weights released with commercial friendly license

1100+ researchers,
engineers, lawyers, and
policy makers



The Stack

Data collection



Find the filtered and deduplicated datasets at: www.hf.co/bigcode

Data inspection + Opt-out

 **BigCode**

The Stack is an open governance interface between the AI community and the open source community.

Am I in The Stack?

As part of the BigCode project, we released and maintain [The Stack](#), a 3.1 TB dataset of permissively licensed source code in 30 programming languages. One of our goals in this project is to give people agency over their source code by letting them decide whether or not it should be used to develop and evaluate machine learning models, as we acknowledge that not all developers may wish to have their data used for that purpose.

This tool lets you check if a repository under a given username is part of The Stack dataset. Would you like to have your data removed from future versions of The Stack? You can opt-out following the instructions [here](#).

The Stack version:

v1.1

Your GitHub username:

Check!

StarCoder

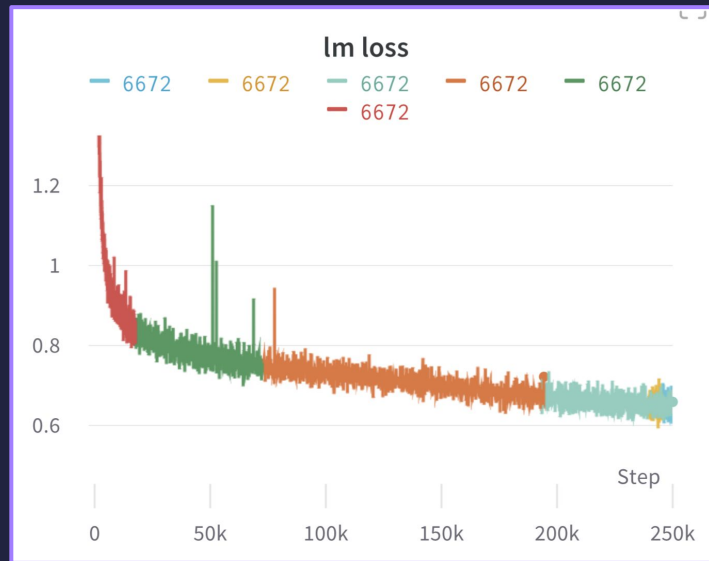
Model size: 15B parameters

Context length: 8096 tokens

Infrastructure: 512 A100 GPUs

Training length: 1T tokens / 250k steps

Training time: 24 days

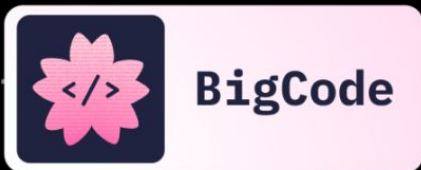


"smooth sailing"

Best open LLM for code at the time of release!



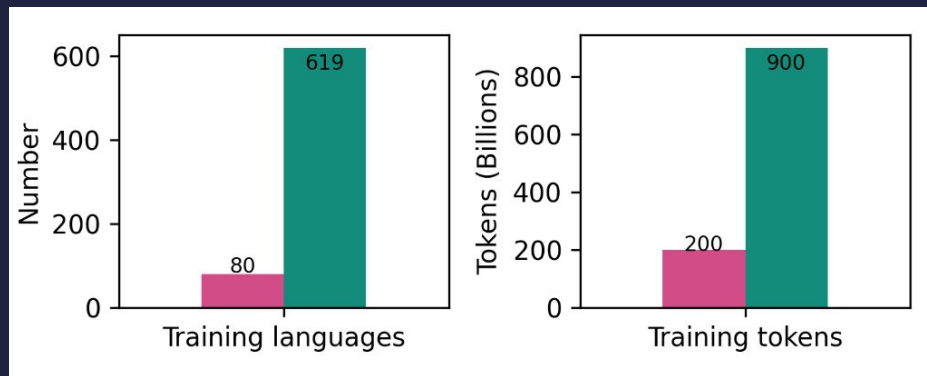
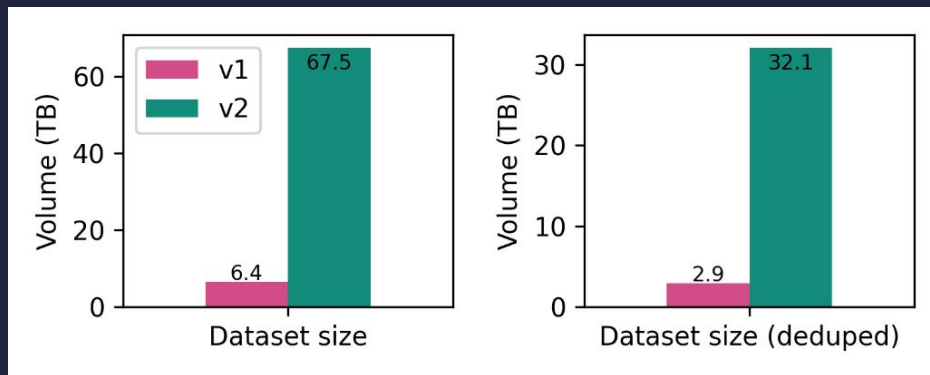
The Stack v2



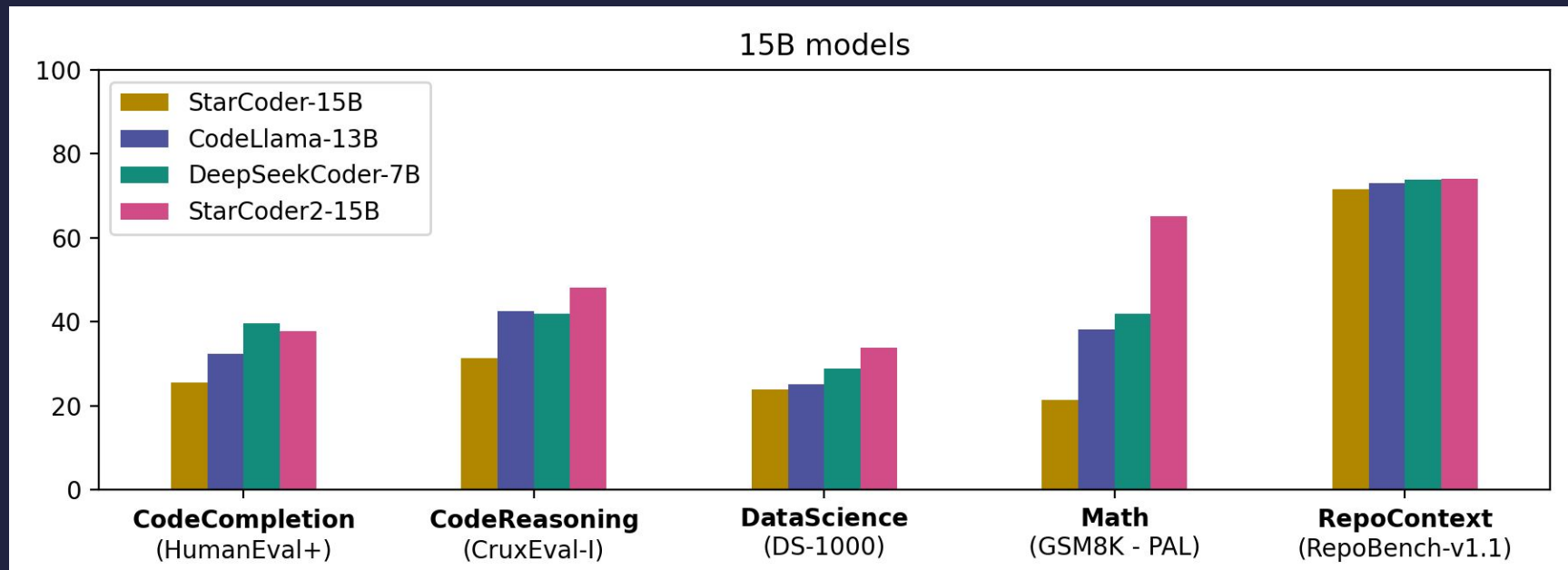
×



The Stack v2



StarCoder2



StarChat2

Write a function that integrates any python function $f(x)$ on an interval $[x_start, x_end]$ using simpsons rule

- Sure! I can help you with that. Here's the Python code for a function that implements Simpson's Rule to integrate a given Python function 'f' over an interval '[x_start, x_end]':

```
import math

def simpsons_rule(f, x_start, x_end, n):
    h = (x_end - x_start) / n

    # Calculate the sum of odd terms and even terms separately
    odd_sum = 0
    even_sum = 0

    for i in range(1, n // 2 + 1):
        xi = x_start + (i * 2 - 1) * h
        odd_sum += f(xi)

    for j in range(1, n // 2):
        xj = x_start + j * 2 * h
        even_sum += f(xj)

    integral = (h / 3) * (f(x_start) + 4 * odd_sum + 2 * even_sum + f(x_end))

    return integral
```

You can call this function by providing your own Python function as 'f', the start value ('x_start'), end value ('x_end'), and the number of subintervals ('n'). The function will calculate and return the approximate definite integral using Simpson's Rule.

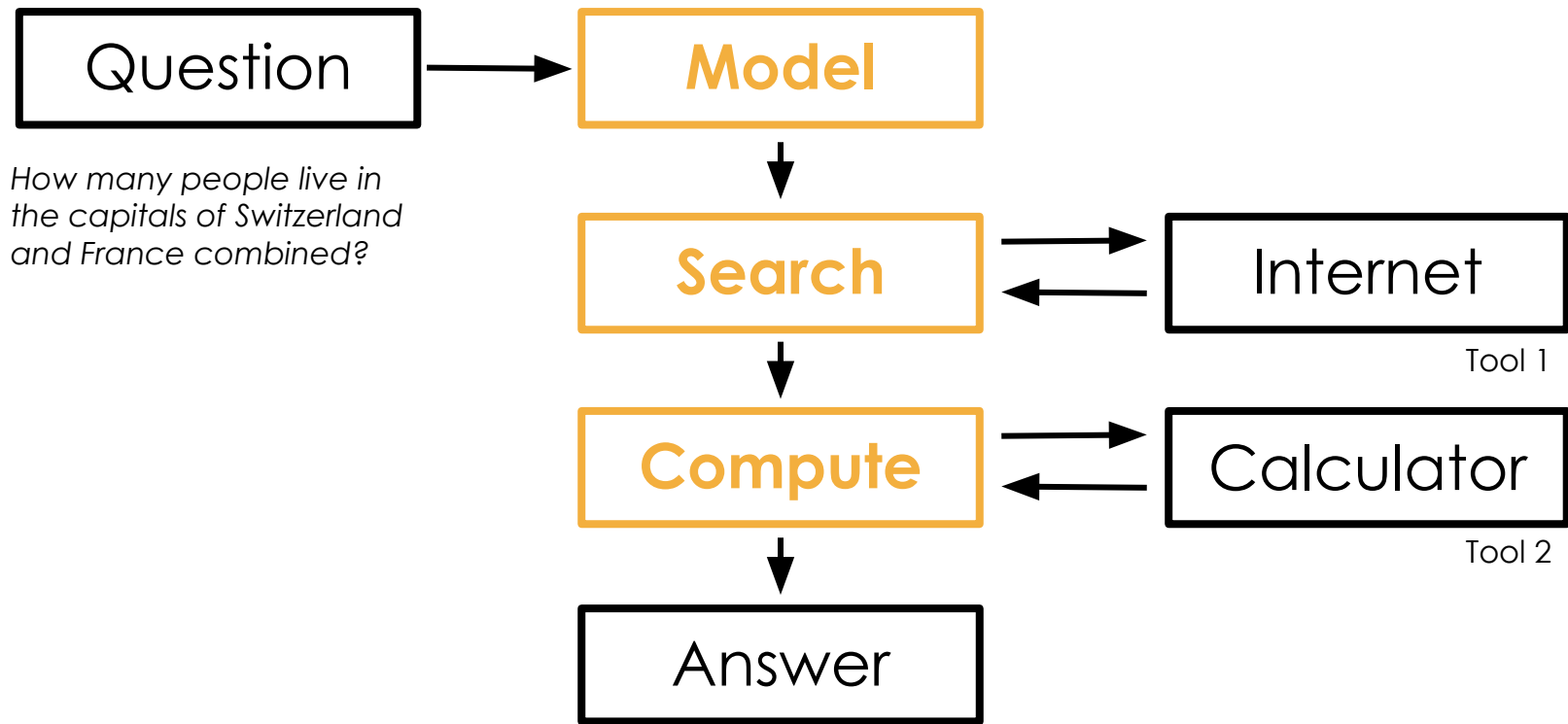
Let me know if you need further assistance or explanation about the implementation.

Trends: **agents & tools**



*How many people live in
the capitals of Switzerland
and France combined?*

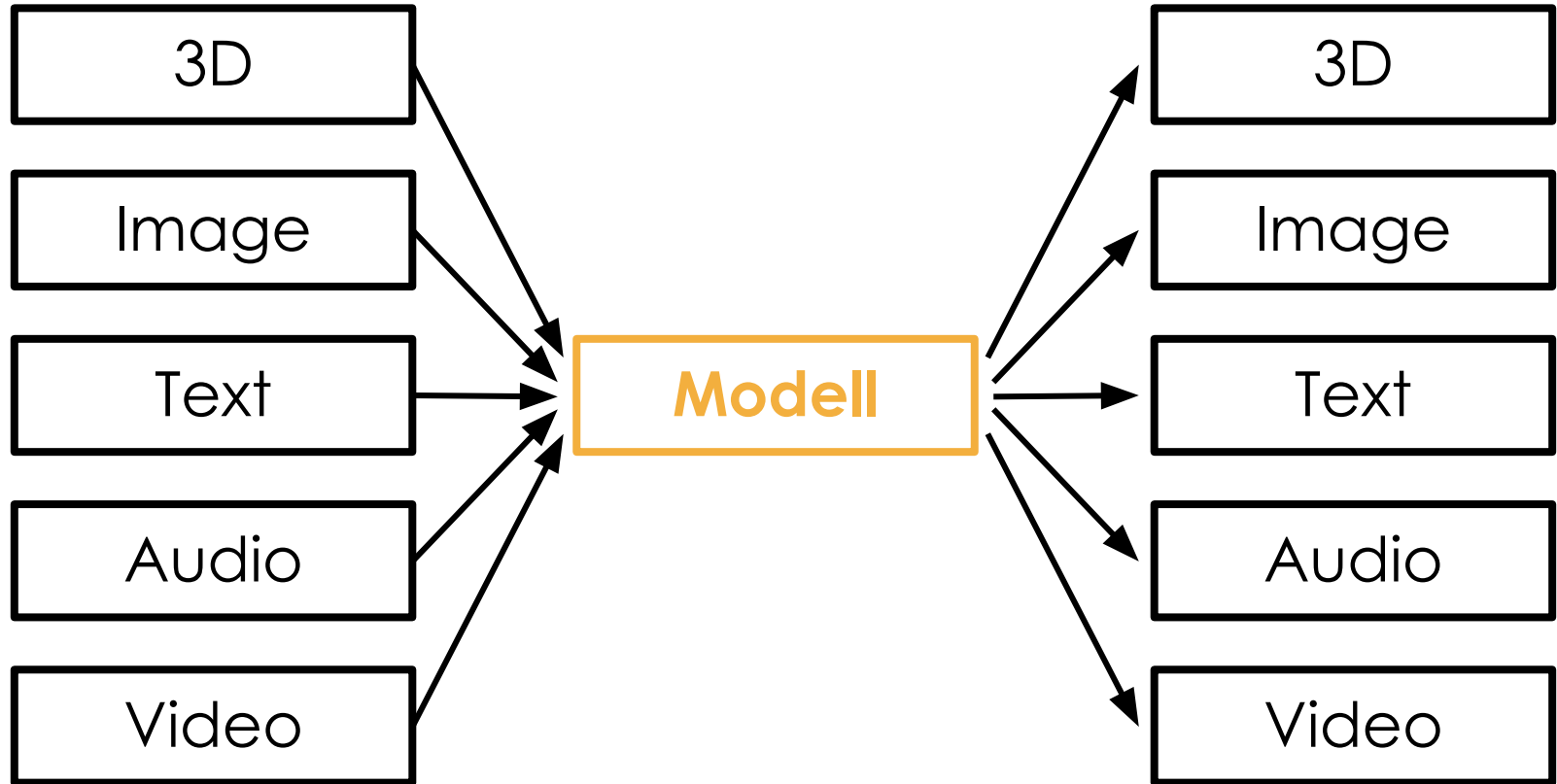
Trends: **agents & tools**



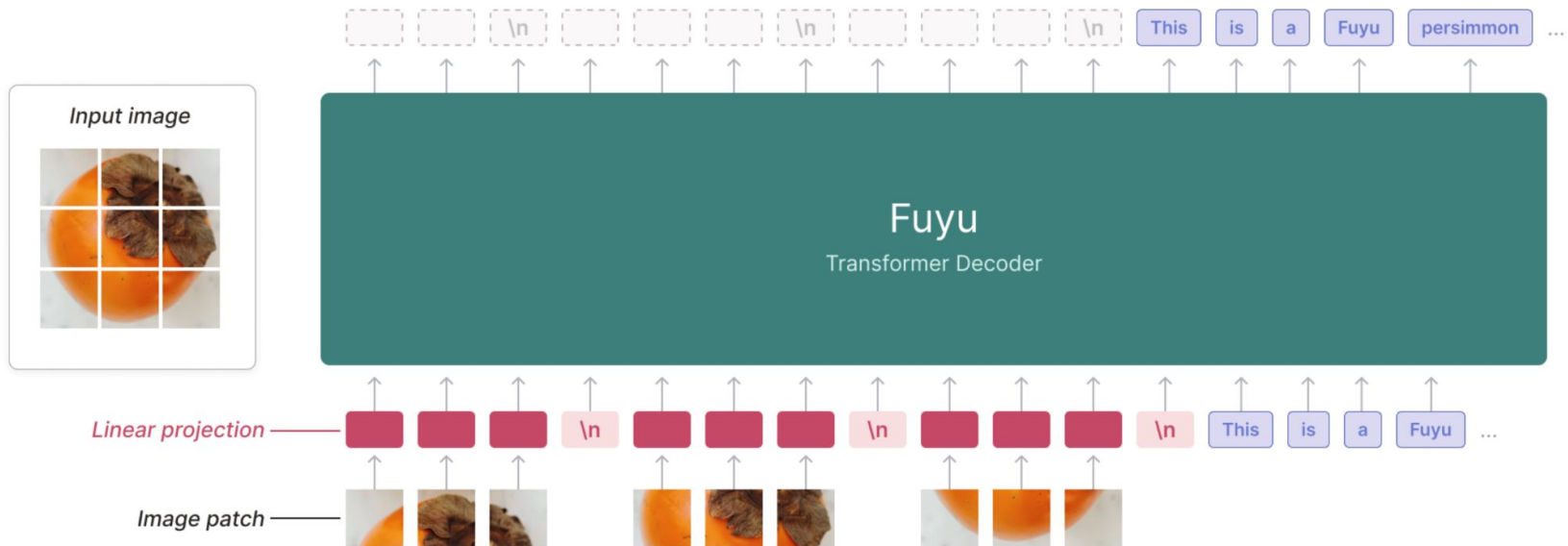
Trends: **multimodality**



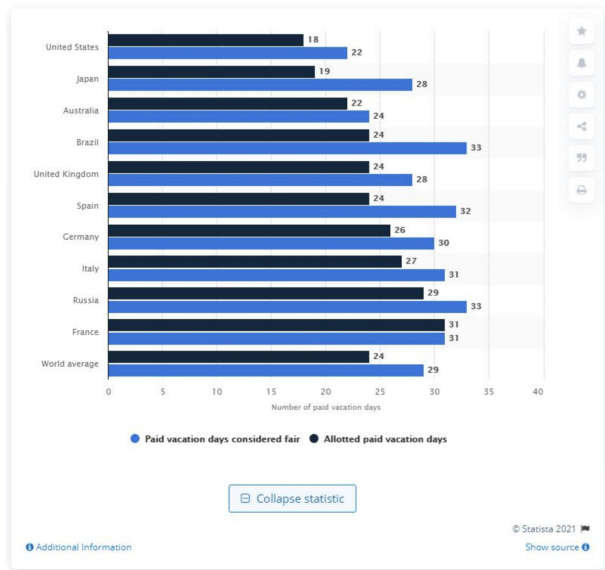
Trends: **multimodality**



Example: Fuyu



Example: Fuyu



Question: “What was the fair amount of paid vacation days in the UK?”

Fuyu's answer: “28”

Questions?

GitHub/HF Hub/X: [lvwerra](#)