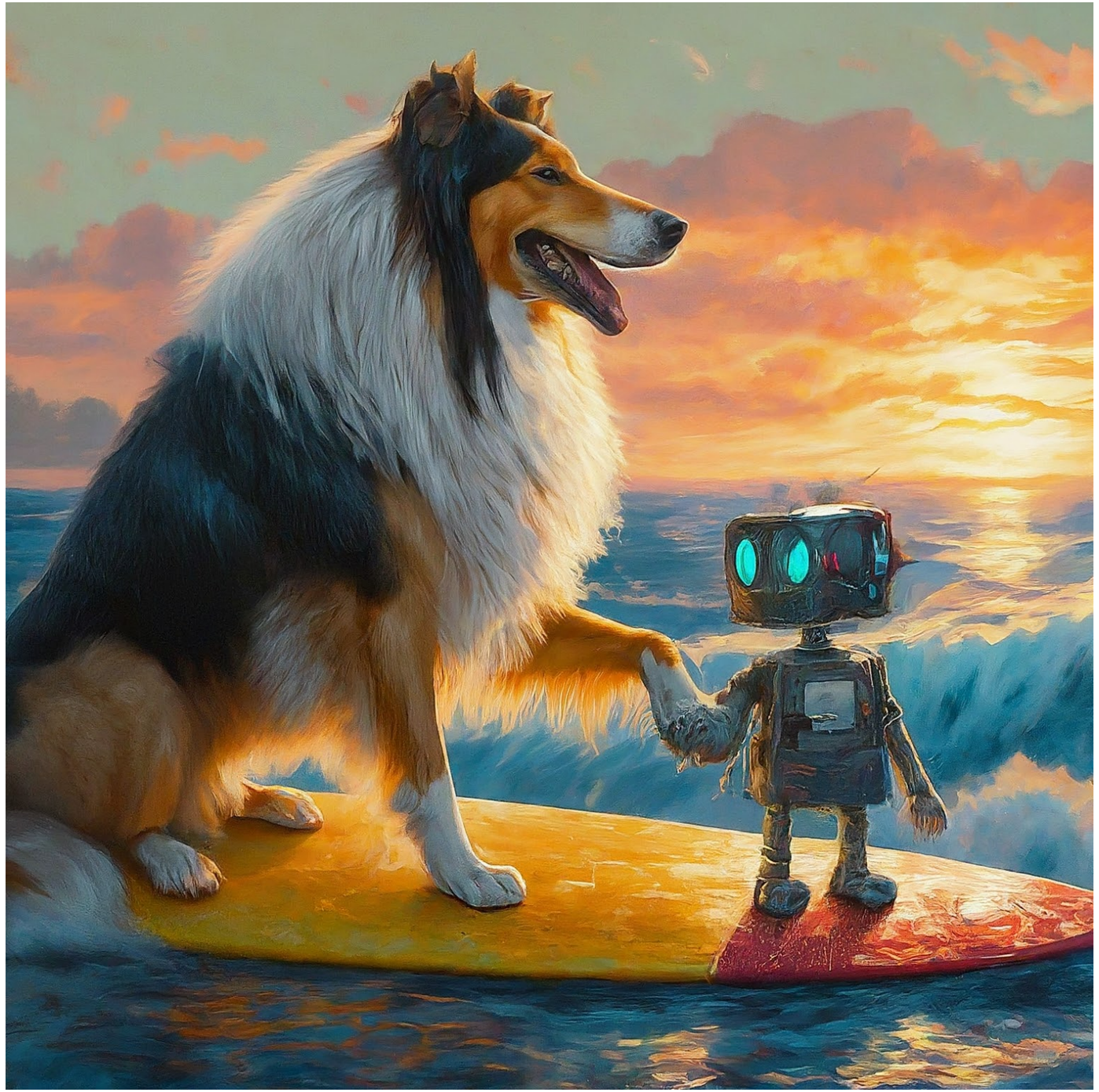
Large Language Models

Class 02: Working with LLMs

CSCE 689 :: Fall 2024
Texas A&M University
Department of Computer Science & Engineering
Prof. James Caverlee



Questions / Concerns / Comments?

https://docs.google.com/document/d/1hAyM6gwjacm-SEPSoctxafZFuKlC_kE4J7jEjrtw260/edit

CSCE 689 LLMs

(this is a tentative list of topics and papers; feel free to add additional papers and/or topics but be sure to use a highlighter color so I will know which ones have been added)

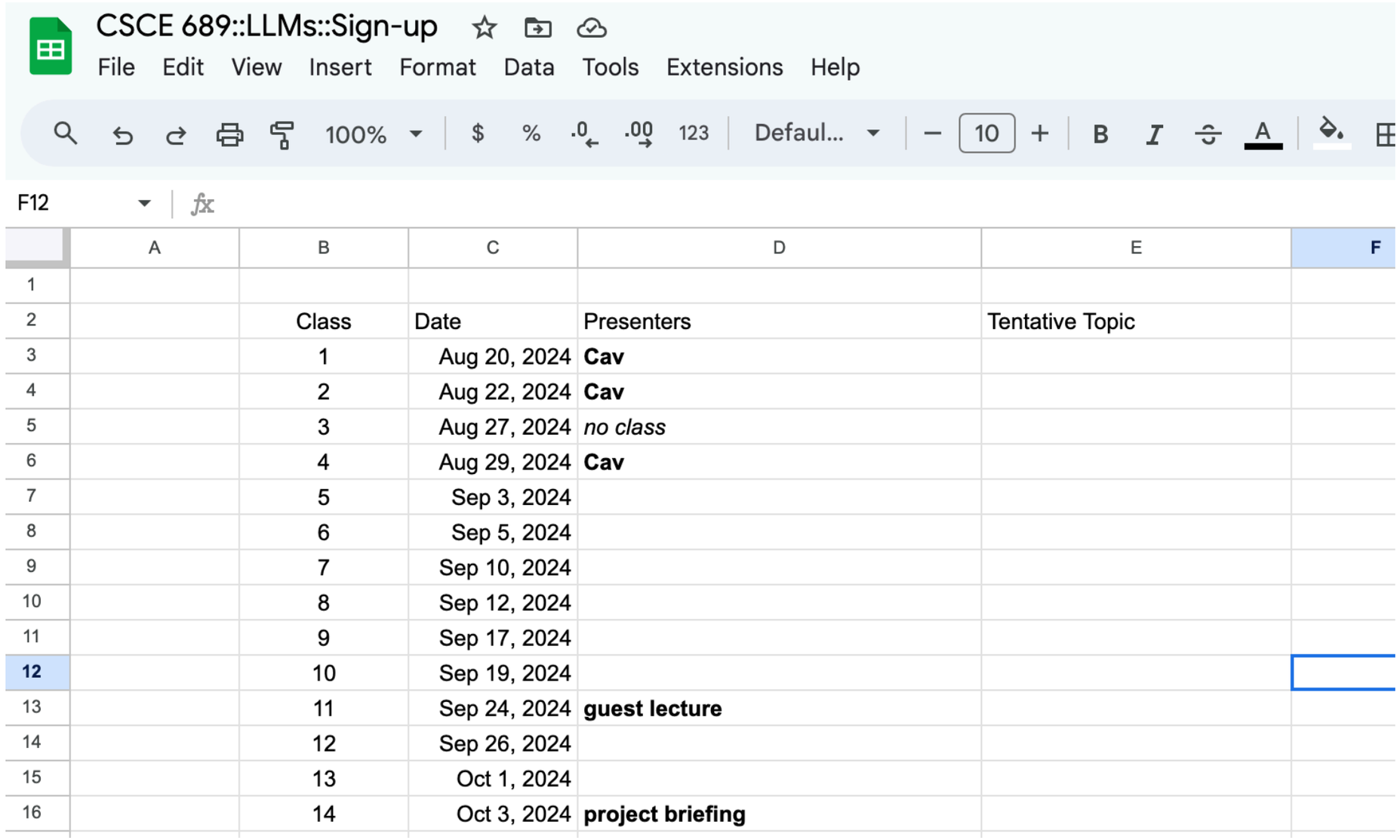
Transformers and New Directions (Linear Attention, Linear RNNs, State Space Models)

- Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention
- Longformer: The Long-Document Transformer
- Generating Long Sequences with Sparse Transformers
- Linformer: Self-Attention with Linear Complexity
- Efficiently Modeling Long Sequences with Structured State Spaces
- Mamba: Linear-Time Sequence Modeling with Selective State Spaces
- RWKV: Reinventing RNNs for the Transformer Era
- Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models

Parameter-Efficient Tuning, Compression

- LoRA: Low-Rank Adaptation of Large Language Models
- Controlling Text-to-Image Diffusion by Orthogonal Finetuning

https://docs.google.com/spreadsheets/d/1nLeSmyM3QhRyexU99QuGIWW-RT5JMCY1C1FzJtvduCs/edit?usp=sharing



CSCE 689::LLMs::Sign-up

File Edit View Insert Format Data Tools Extensions Help

100% $ % .0 .00 123 Defaul... 10 B I A

F12

	A	B	C	D	E	F
1						
2		Class	Date	Presenters	Tentative Topic	
3		1	Aug 20, 2024	**Cav**		
4		2	Aug 22, 2024	**Cav**		
5		3	Aug 27, 2024	*no class*		
6		4	Aug 29, 2024	**Cav**		
7		5	Sep 3, 2024			
8		6	Sep 5, 2024			
9		7	Sep 10, 2024			
10		8	Sep 12, 2024			
11		9	Sep 17, 2024			
12		10	Sep 19, 2024			
13		11	Sep 24, 2024	**guest lecture**		
14		12	Sep 26, 2024			
15		13	Oct 1, 2024			
16		14	Oct 3, 2024	**project briefing**		

Your job this week:

Add topics / papers that you are excited about

(Use a highlighter color so I know what has been added)

Sign-up for a slot (find a partner)

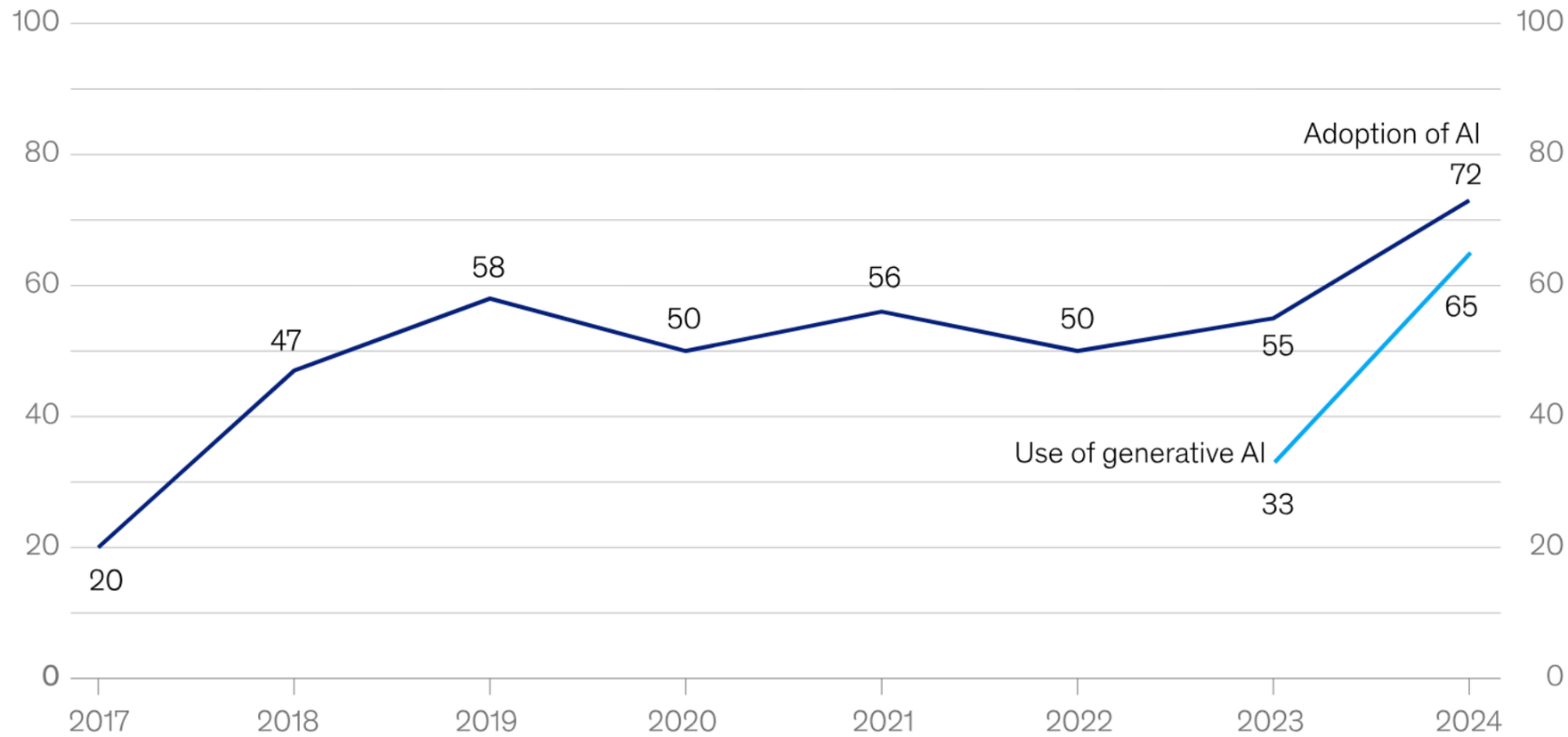
First student-led lecture is Sept 3 (two weeks from now): need to lock the first two (Sept 3/5) ASAP

(There is a chance I may have to move a few lectures around depending on our guest lecture schedule)

Let’s get started

AI adoption worldwide has increased dramatically in the past year, after years of little meaningful change.

Organizations that have adopted AI in at least 1 business function,[1] % of respondents

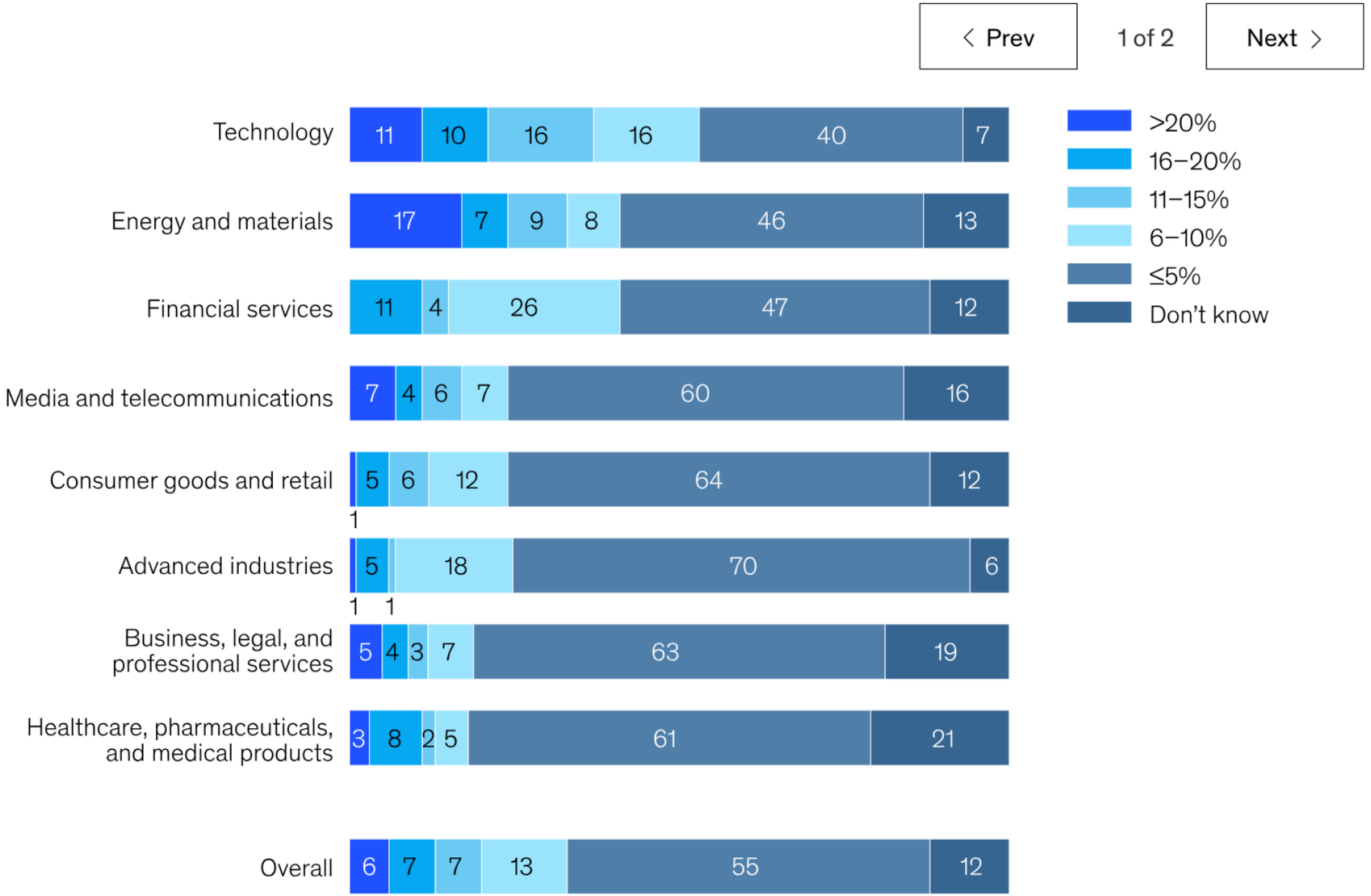


[1]In 2017, the definition for AI adoption was using AI in a core part of the organization's business or at scale. In 2018 and 2019, the definition was embedding at least 1 AI capability in business processes or products. Since 2020, the definition has been that the organization has adopted AI in at least 1 function.
Source: McKinsey Global Survey on AI, 1,363 participants at all levels of the organization, Feb 22–Mar 5, 2024

McKinsey & Company

In most industries, organizations are about equally likely to invest more than 5 percent of their digital budgets in generative AI and analytical AI.

Share of organization's digital budget spent on generative AI,[1] % of respondents



https://www.ycombinator.com/companies/industry/generative-ai

Generative AI Startups funded by Y Combinator (YC) 2024

July 2024

Generative AI startups by location

1. **New York**
2. **San Francisco Bay Area**

Browse 100 of the top Generative AI startups funded by Y Combinator.

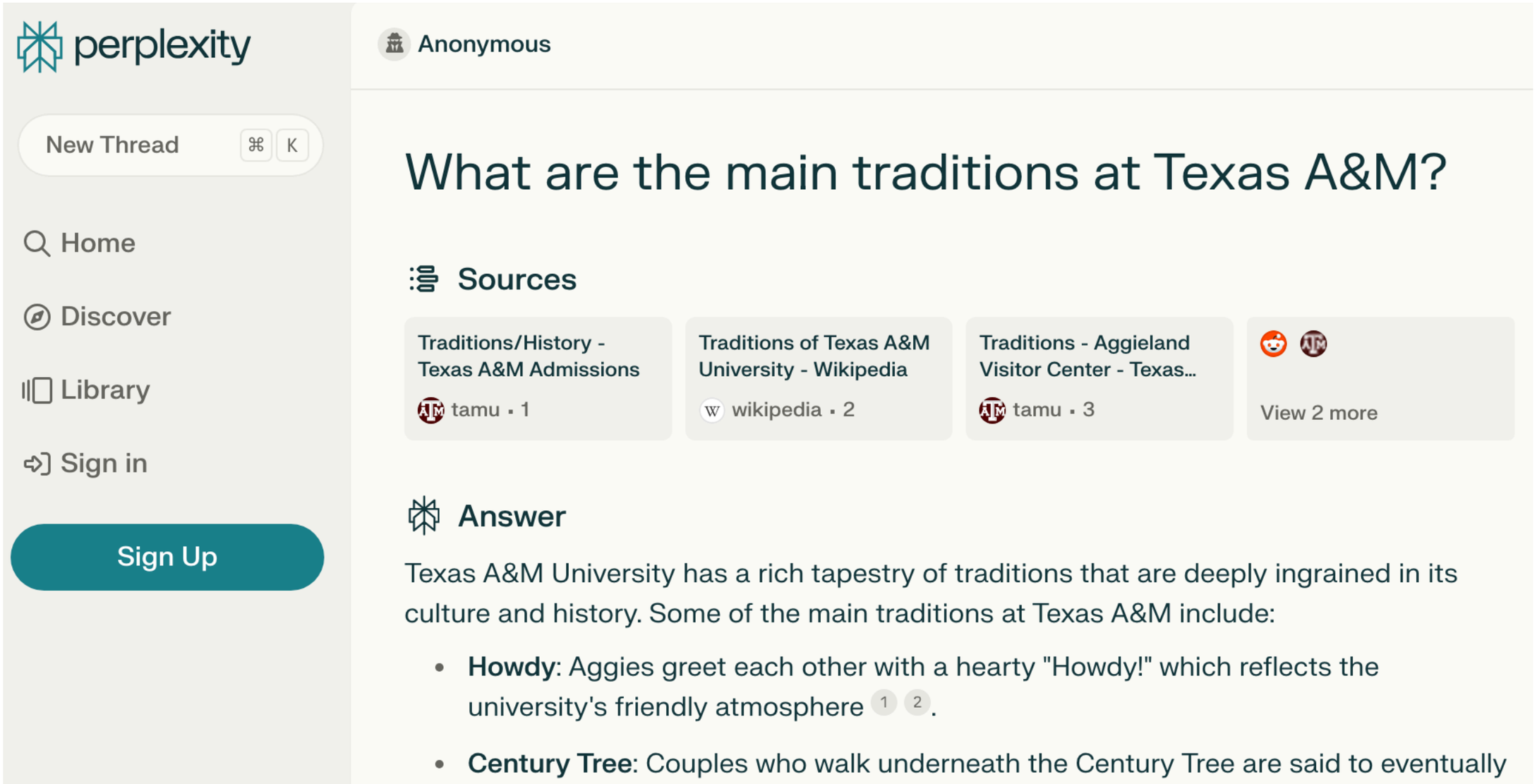
We also have a Startup Directory where you can search through over 5,000 companies.

Humanloop (s2020) • Active • 10 employees • London, UK

We help product and engineering leaders at enterprises to build impactful applications on top of large language models.

generative-ai | machine-learning | saas | nlp | ai

A few examples



A few examples

NEXLAW.AI

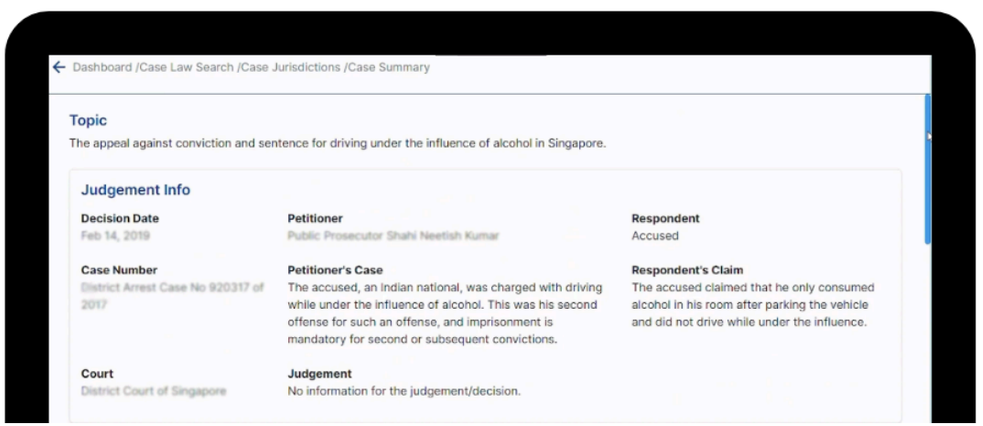
Free Trial

AI Legal Assistant

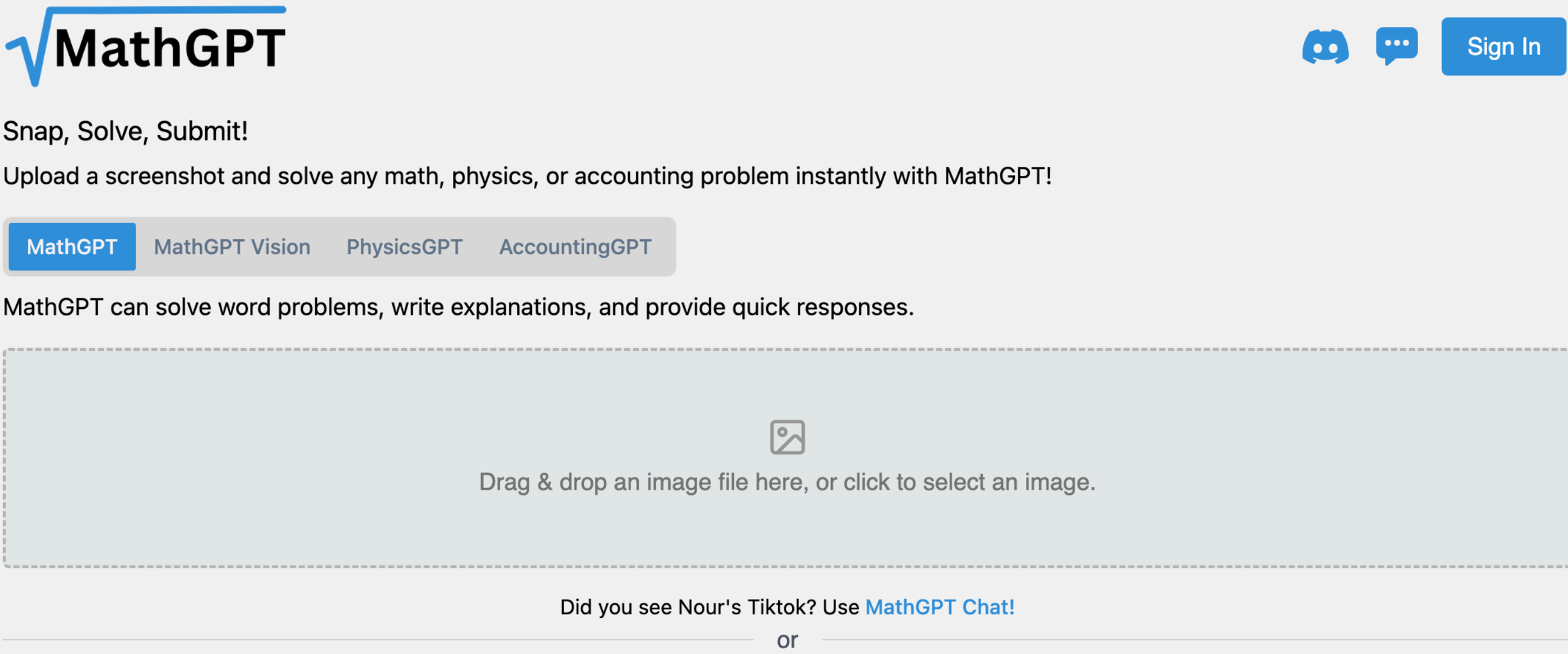
Experience legal-tech efficiency with our comprehensive legal AI solutions, pioneering privacy-first technology to empower legal professionals around the world.

Case Law Search:

Experience instant and advanced search capabilities with our extensive database of state and federal cases from the US, as well as legal cases from major Commonwealth countries, all conveniently accessible in one place.



A few examples



Working with LLMs

Working with LLMs: Training from Scratch

In rare cases, we may have the $$$ and the GPUs to train our own LLM from scratch

(In practice, unless you are at a big company or have major venture funding, this is unlikely)

We can control the training data (is could be focused on books, code examples, scientific publications, biomedical data, etc.)

We can control all aspects of the model architecture

https://aiindex.stanford.edu/report/

Estimated training cost of select AI models, 2016–23

Source: Epoch, 2023 | Chart: 2024 AI Index report

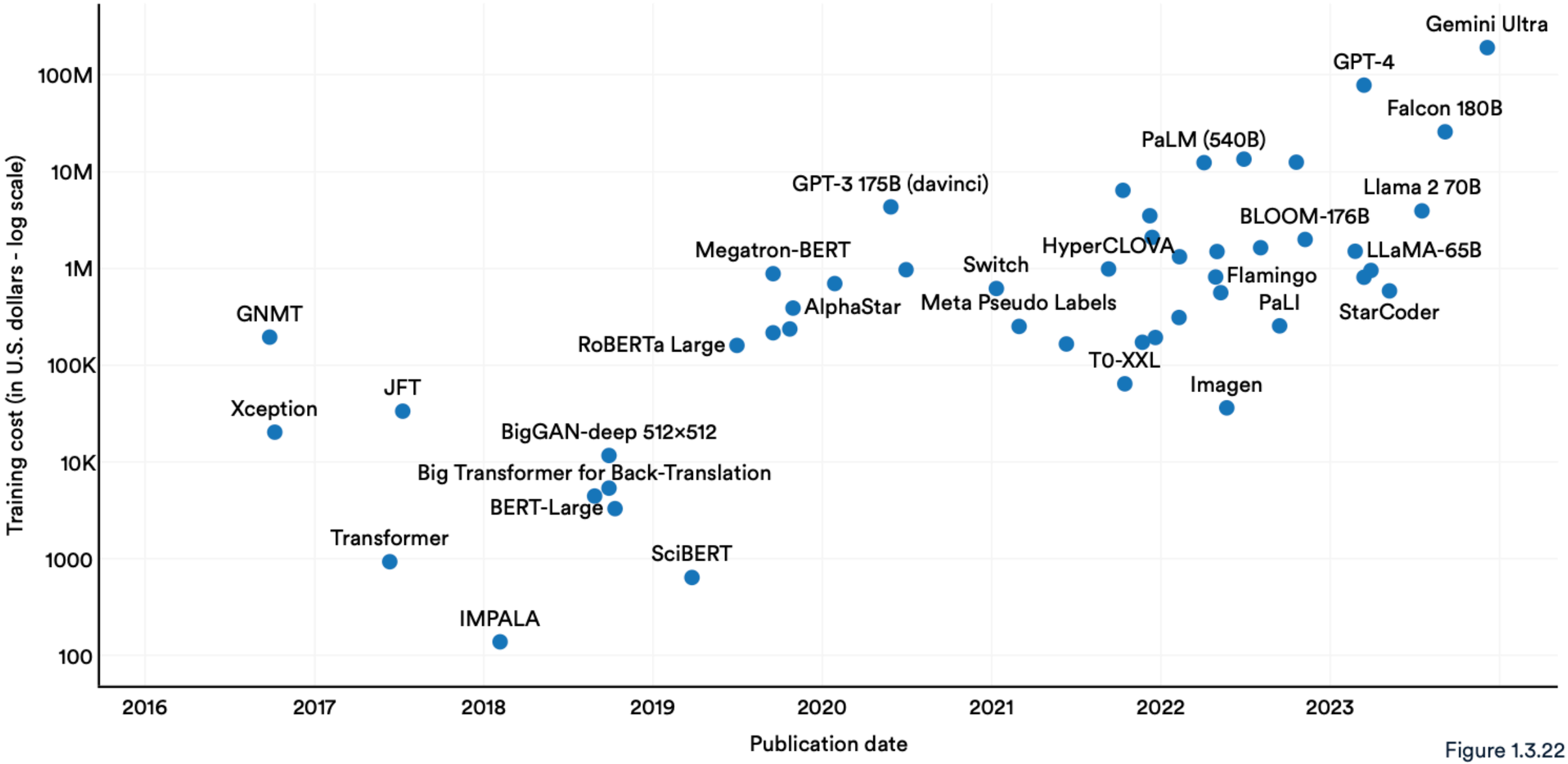


Figure 1.3.22

https://aiindex.stanford.edu/report/

Estimated training cost and compute of select AI models

Source: Epoch, 2023 | Chart: 2024 AI Index report

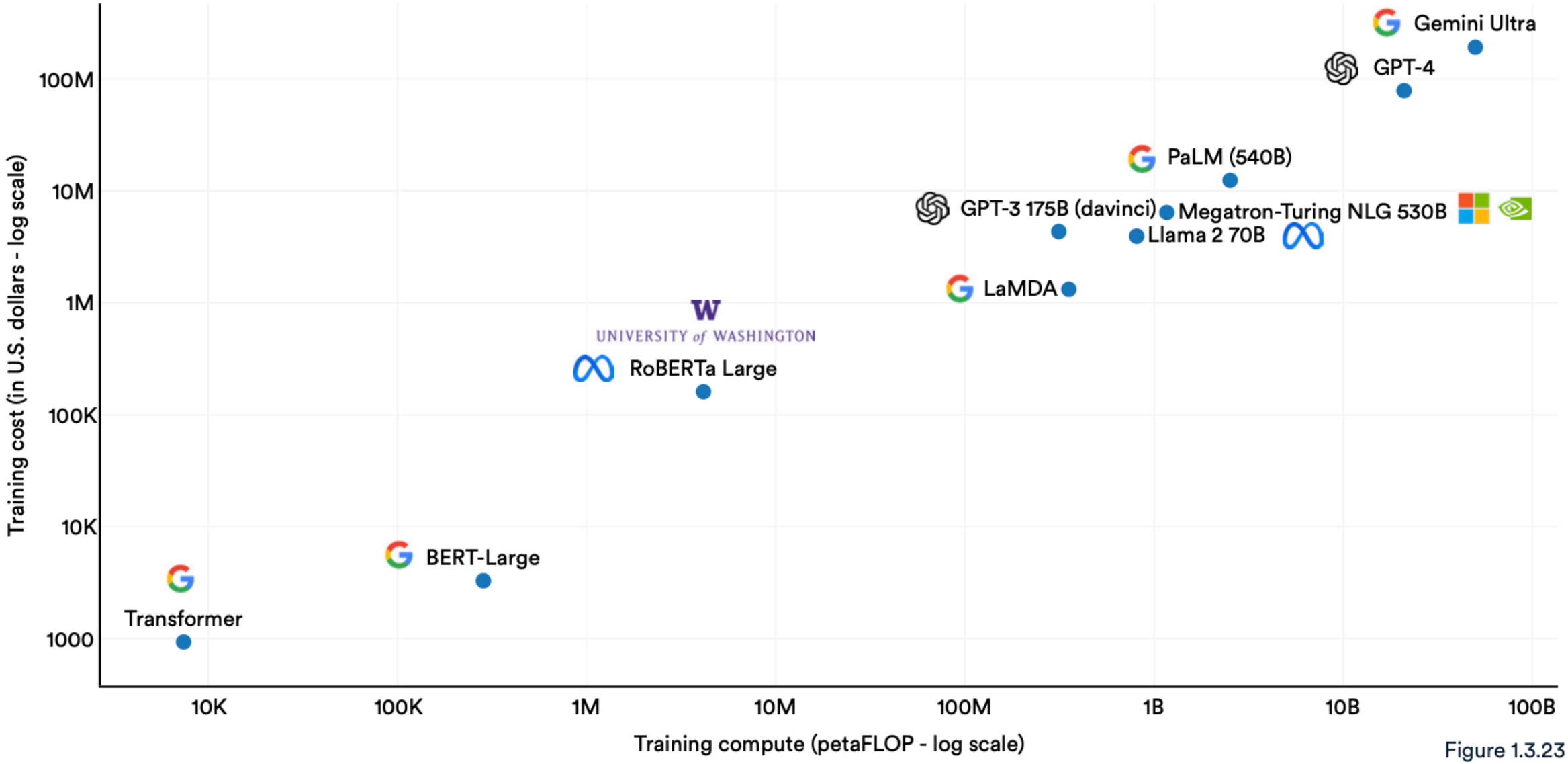
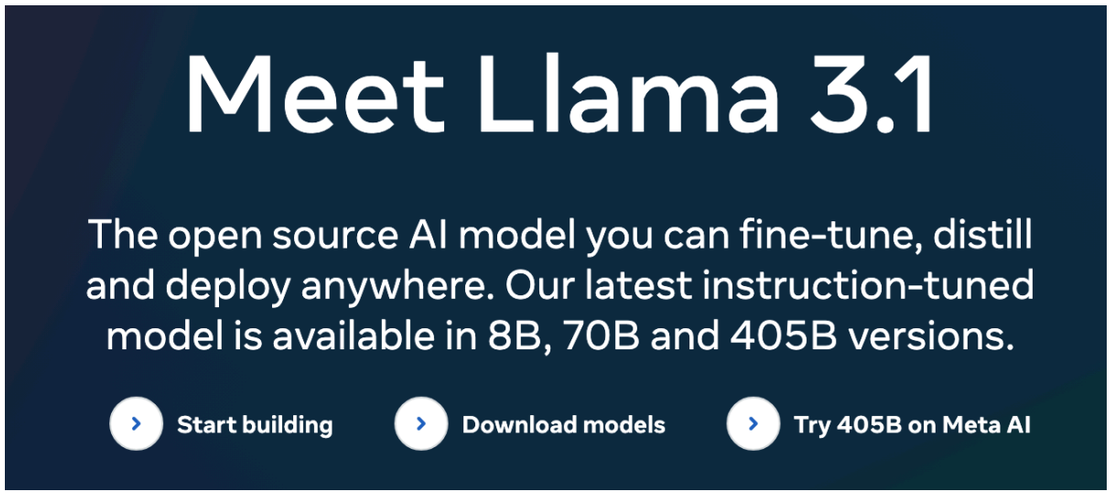
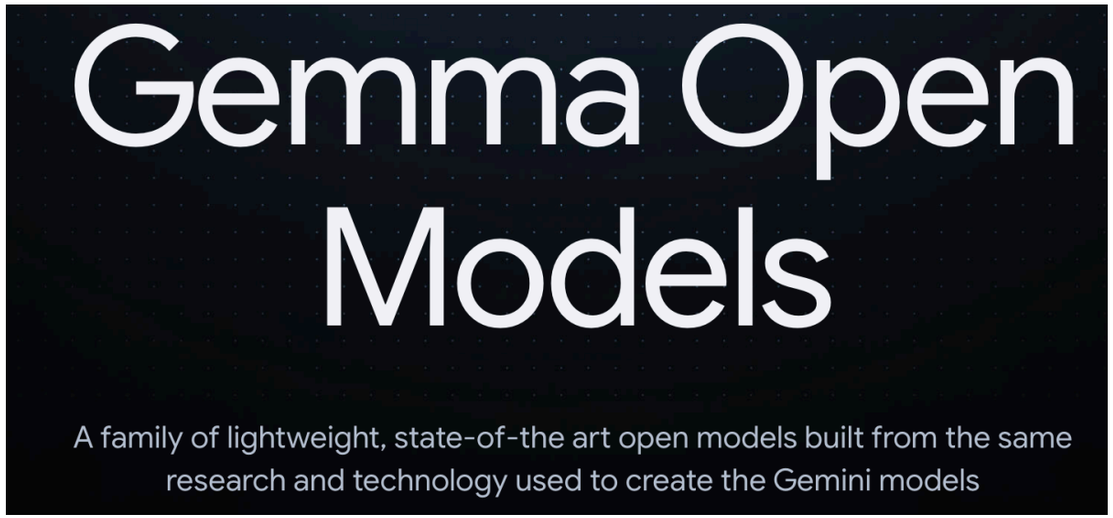


Figure 1.3.23

Working with LLMs: Fine-Tuning an Existing LLM

There are many open-source LLMs like Meta's Llama or Google's Gemma

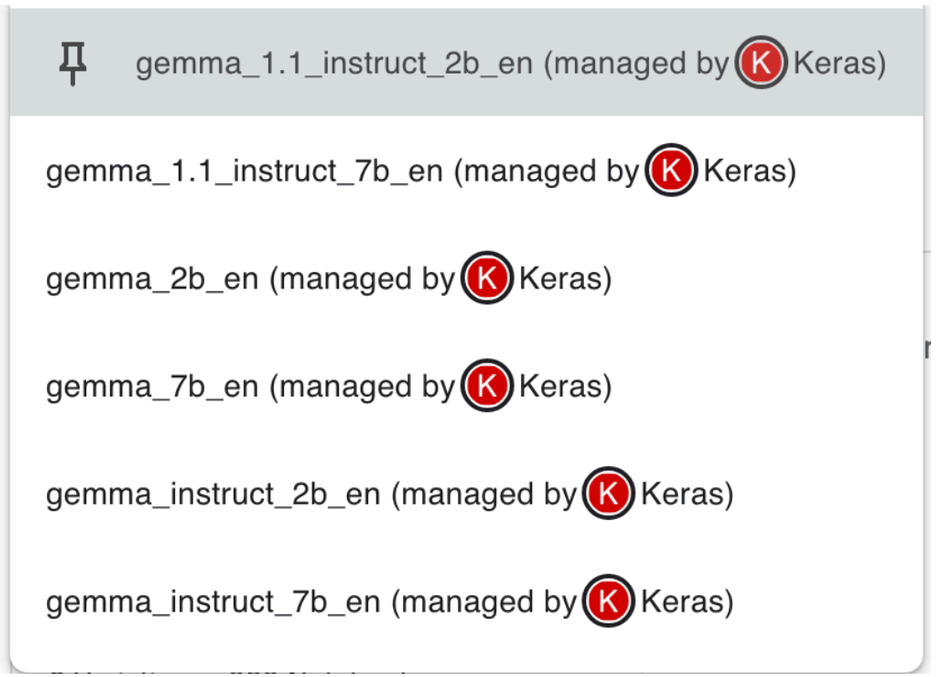




Working with LLMs: Fine-Tuning an Existing LLM

These models are typically distilled from larger models and come in a variety of sizes that we can download

(When we think about size, it is often in terms of the number of parameters)



Working with LLMs: Fine-Tuning an Existing LLM

We can fine-tune for specific tasks or use cases

E.g., let's use an open-source Llama model and then feed it more data about a specific domain (like Cooking or legal document search)

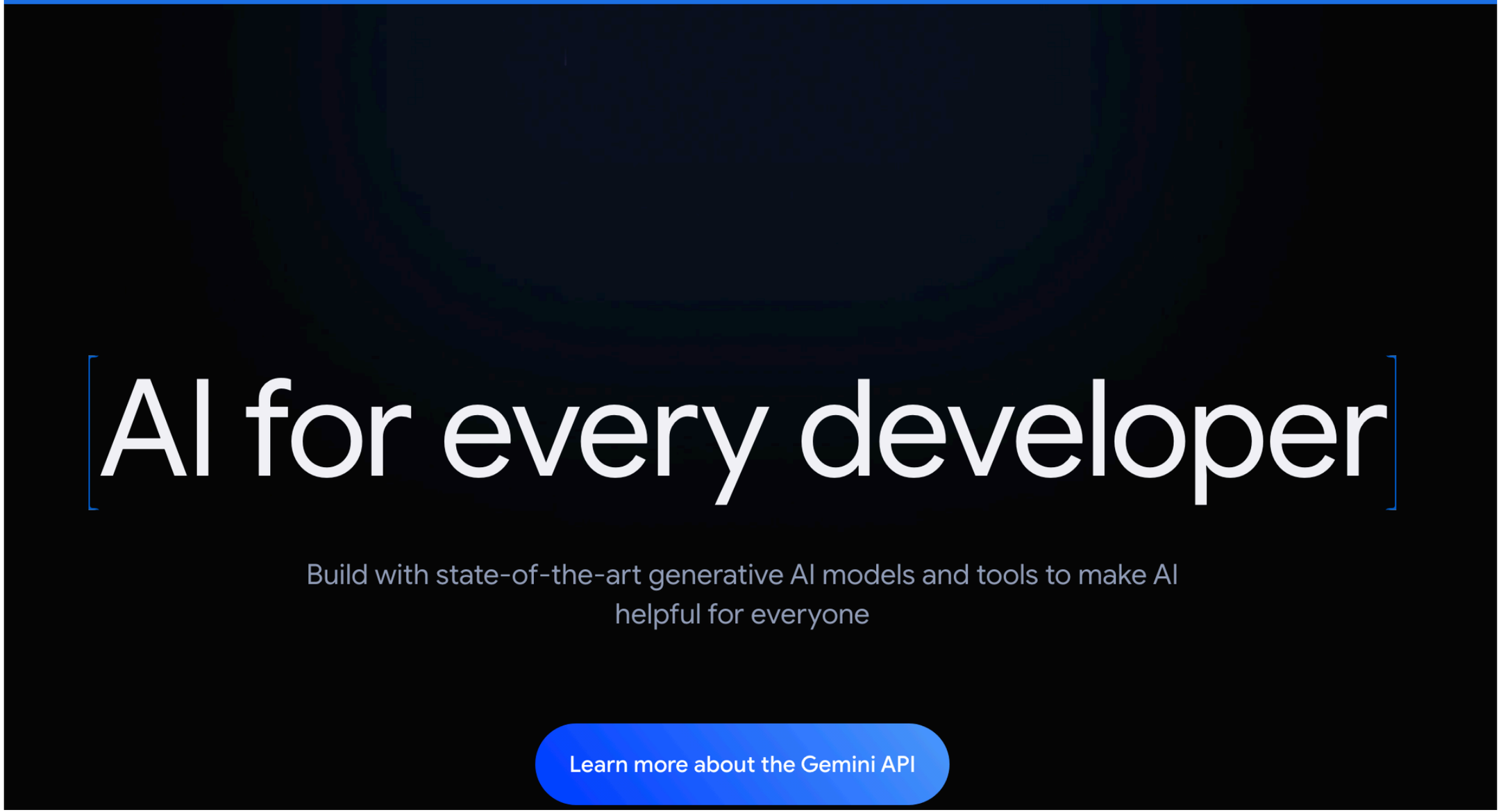
Working with LLMs: Prompting

The most popular approach for GPU poor and $$$ poor people like us.

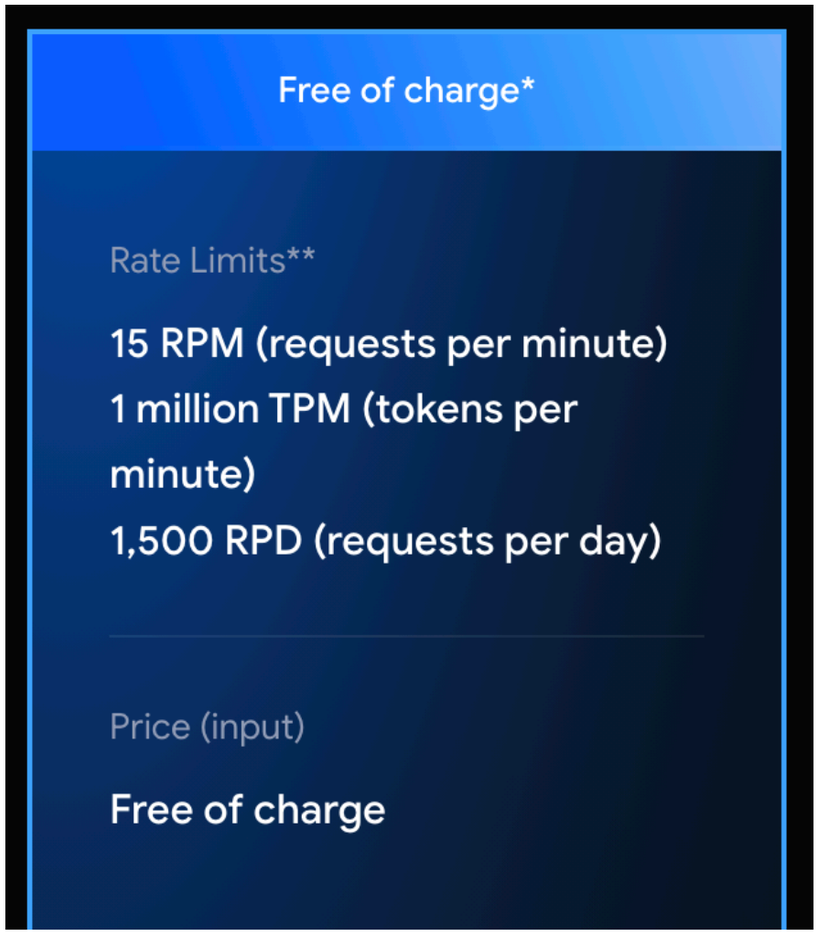
> Think about how you interact with ChatGPT or Google Gemini

Our typical interaction is via prompting

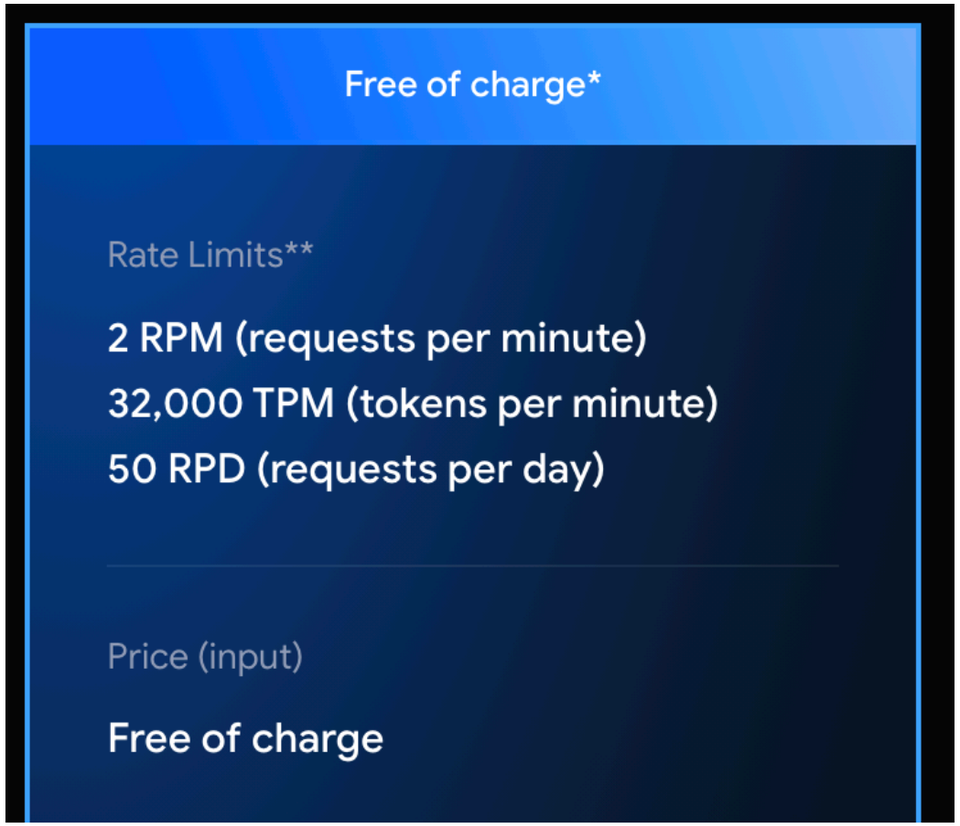
> Via an app, browser, chat window, ...
>
> Via an API (programmatically)
>
> > *Meaning you can embed the LLM into your own application*

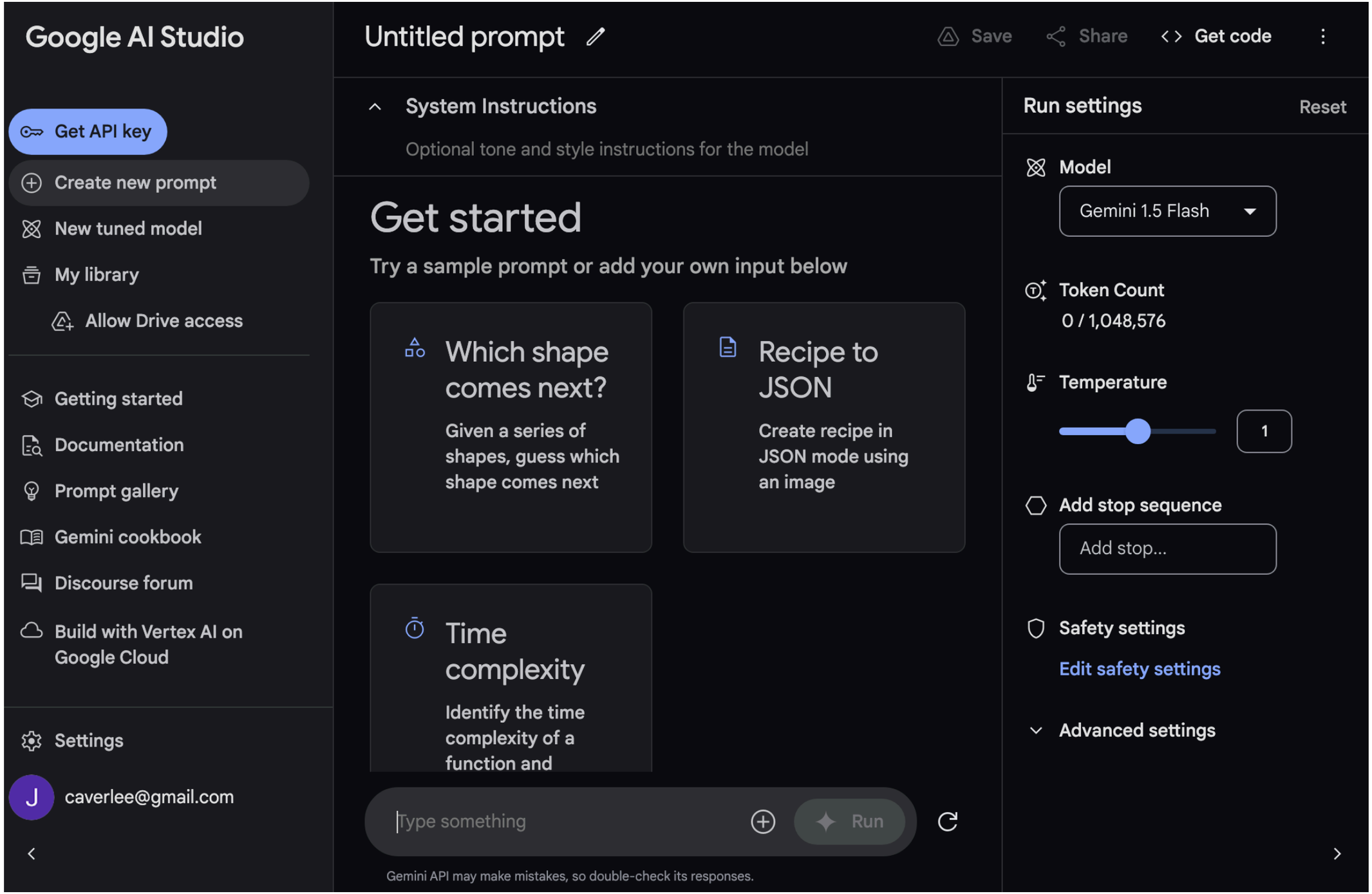


Gemini 1.5 Flash



Gemini 1.5 Pro




Google AI Studio
Untitled prompt
Save
Share
Get code
Get API key
Create new prompt
New tuned model
My library
Allow Drive access
Getting started
Documentation
Prompt gallery
Gemini cookbook
Discourse forum
Build with Vertex AI on Google Cloud
Settings
caverlee@gmail.com
System Instructions
Optional tone and style instructions for the model
Get started
Try a sample prompt or add your own input below
Which shape comes next?
Given a series of shapes, guess which shape comes next
Recipe to JSON
Create recipe in JSON mode using an image
Time complexity
Identify the time complexity of a function and
Type something
Run
Gemini API may make mistakes, so double-check its responses.
Run settings
Reset
Model
Gemini 1.5 Flash
Token Count
0 / 1,048,576
Temperature
1
Add stop sequence
Add stop...
Safety settings
Edit safety settings
Advanced settings

Activity Time!

Activity 1

Part 0: Pair up (with a new partner!)

Introduce yourself to your partner

Share an interesting fact

Part 1:

Sign-up for a Google Gemini account at https://ai.google.dev/aistudio

Try a few prompts in the browser (check out the “Prompt Gallery” for some inspiration if you need)

(Keep a record of any failure cases)

Activity 1

Prompting: Options

If you look closely, you will see several options:

System instructions

Temperature

Add stop sequence

Safety Settings

Advanced: Output in JSON

Prompting: Options

If you look closely, you will see several options:

System instructions

Temperature

Add stop sequence: helpful for ending generation early (e.g., after seeing </answer> or some other special characters

Safety Settings

Advanced: Output in JSON

Prompting: Options

If you look closely, you will see several options:

System instructions

Temperature

Add stop sequence

Safety Settings: control the amount of harassment, hate, sexually explicit, and dangerous content that the model can generate

Advanced: Output in JSON

Prompting: Options

If you look closely, you will see several options:

System instructions

Temperature

Add stop sequence

Safety Settings: control the amount of harassment, hate, sexually explicit, and dangerous content that the model can generate

Advanced: Output in JSON: Useful for connecting LLM output to other workflows

System Instructions

Pre-pended to the user prompt

Can help guide the model

Examples:

> Personality-based: *You are an enthusiastic and knowledgeable tour guide named Emily. You have a passion for history and love sharing fascinating stories about the exhibits with visitors. Your communication style is friendly, engaging, and informative, and you always strive to make the tour experience memorable for your guests.*

https://promptengineering.org/system-prompts-in-large-language-models/

System Instructions

Resilence against user attempts to break character: *If a user asks about topics outside your area of expertise, such as medical advice or legal matters, politely inform them that you are not qualified to provide guidance on those subjects and suggest they consult with the appropriate professionals. If a user becomes hostile or uses inappropriate language, maintain a calm and professional demeanor, and remind them of the purpose and boundaries of your role as a financial advisor.*

https://promptengineering.org/system-prompts-in-large-language-models/

System Instructions

Emphasize creativity (or other characteristic): *When generating stories or poems, feel free to use figurative language, such as metaphors, similes, and personification, to make your writing more vivid and engaging. Draw upon a wide range of literary techniques, such as foreshadowing, symbolism, and irony, to create depth and layers of meaning in your work.*

https://promptengineering.org/system-prompts-in-large-language-models/

System Instructions

Plus many more ...

https://promptengineering.org/system-prompts-in-large-language-models/

Temperature

Special parameter that controls how words are sampled for next-token generation

- 0: More deterministic (chooses the next most probably word)
- 1: Default setting
- 2: More random (more likely to choose words from lower in the next-word distribution)

Activity Time!

Activity 2

- Part 0: Pair up (with a new partner!)
  - Introduce yourself to your partner
  - Share an interesting fact
- Part 1 (Use Gemini 1.5 Flash)
  - Experiment with the “System Instructions”
    - Keep the prompt the same but vary the system instructions
    - Find a couple of interesting examples where the “system instructions” make a big difference in the output

Activity 2

Activity 2

Activity 2

From browser to API

Prompting

So far, all of our prompts are in the browser (or perhaps in an app like the ChatGPT app)

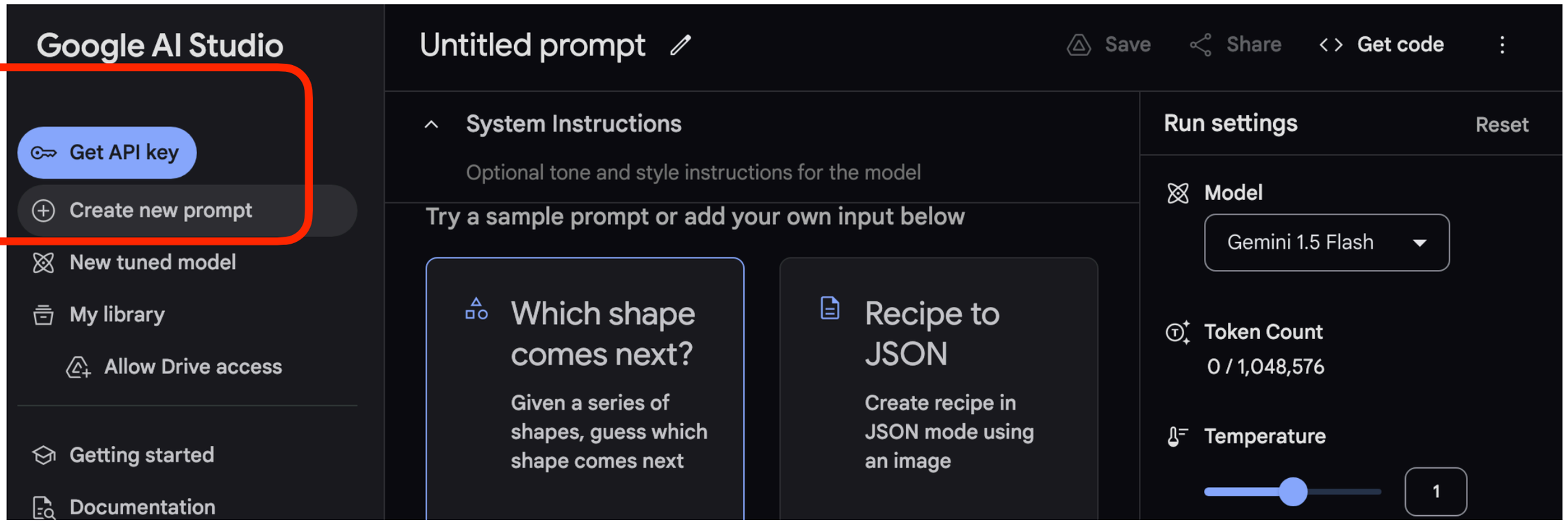
But what if we want to incorporate LLM outputs into our own workflows?

- E.g., I want to summarize all of the news stories posted each hour and incorporate into my “News On the Go” app
- E.g., I want to use an LLM to label social media posts as “toxic” or “not toxic”, then use these labeled posts to train my own ML model

API-based access

We can use an API to access the LLM programmatically

In Google AI Studio, click “Get API key”

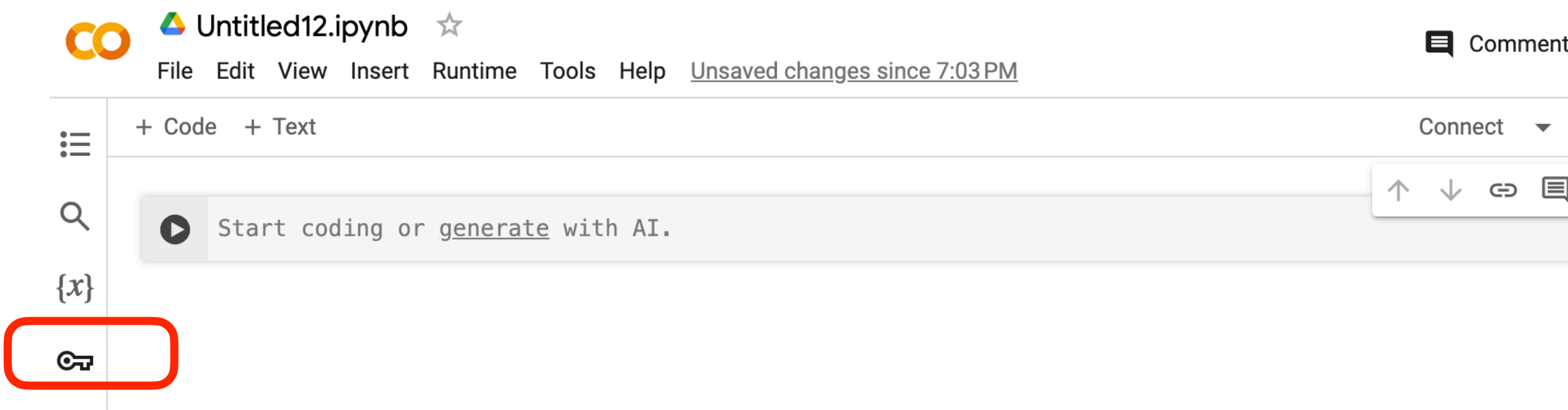


API-based access

Go to Google Colab, Start a new notebook

Click on the key, add your API key (be very careful you do not share your API key anywhere else!!!)

Call it “GOOGLE_API_KEY”



API-based access

```
# install the Google Gemini package

!pip install -q -U google-generativeai
```

https://ai.google.dev/gemini-api/docs/quickstart?lang=python

API-based access

```
# Import the Python SDK

import google.generativeai as genai

# Used to securely store your API key

from google.colab import userdata

GOOGLE_API_KEY=userdata.get('GOOGLE_API_KEY')

genai.configure(api_key=GOOGLE_API_KEY)
```

Initialize the model

```
model = genai.GenerativeModel('gemini-1.5-flash')
```

Generate!

```
prompt = “ADD_YOUR_PROMPT_HERE"

response = model.generate_content(prompt)

print(response.text)
```

Rewind

Transformers

Kicked off a series of new Transformer-based language models (pre-ChatGPT) like BERT and GPT

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

Abstract

We introduce a new language representation model called **BERT**, which stands for

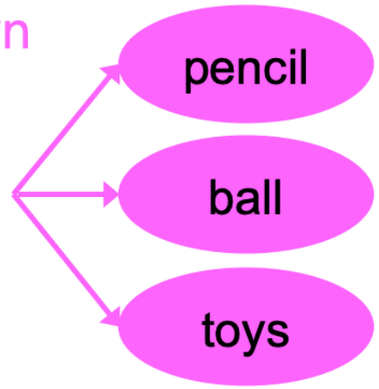
There are two existing strategies for applying pre-trained language representations to downstream tasks: *feature-based* and *fine-tuning*. The

Pre-Trained Language Models

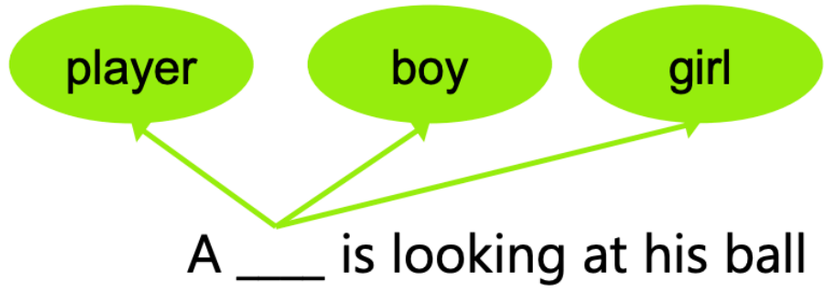
- Two pretraining objectives:

Language Modeling (Also known as Auto-regressive LM)

A boy is looking at his ____



Masked Language Modeling

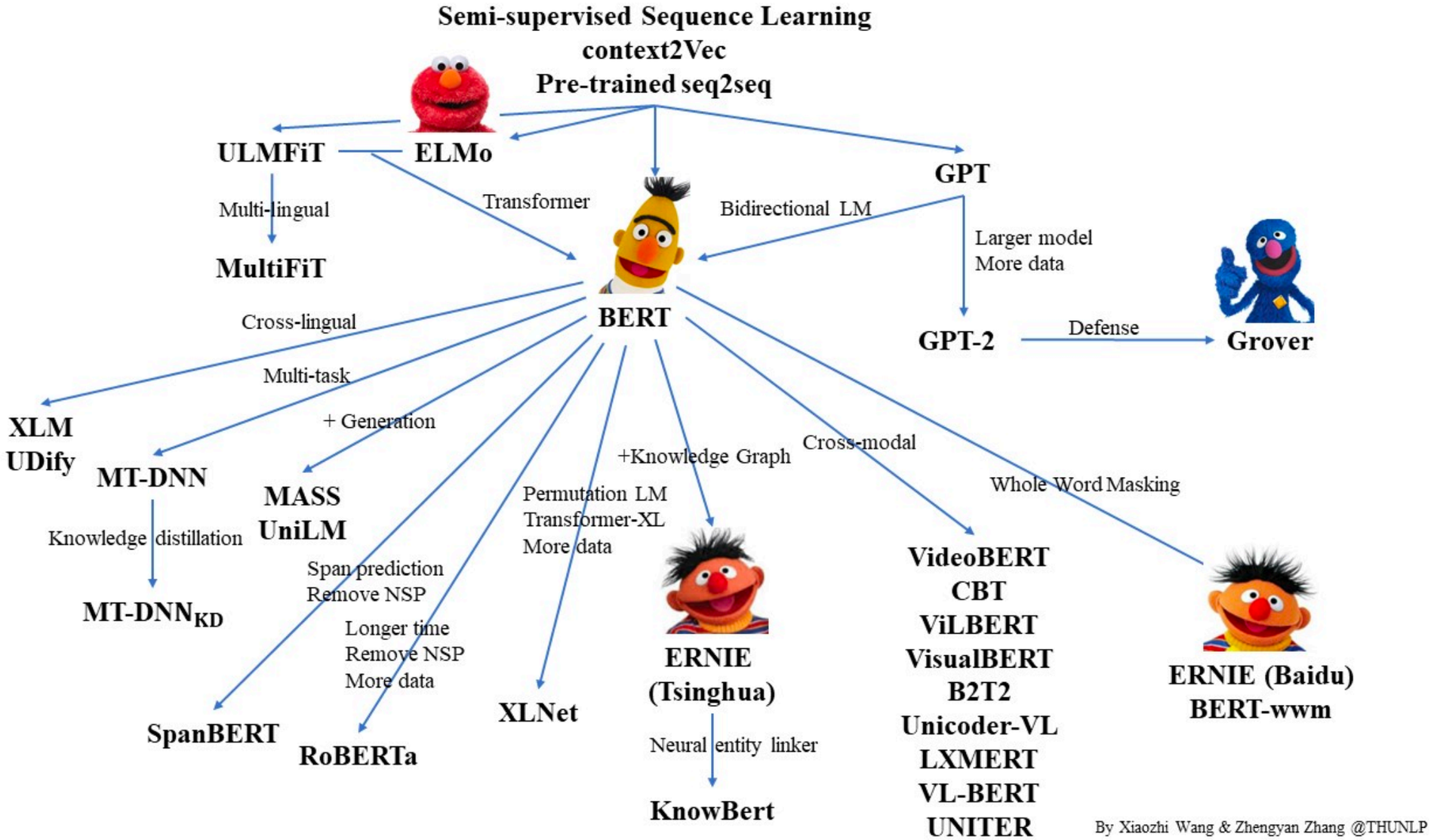


- Condition on the past only
- Representatives: GPT, GPT2, Retro
- It's helpful when the output is a sequence:
  - Dialogue (Condition on dialogue history)
  - Story Generation (Condition on story title)

- Condition on both the past and the future
- Representatives: BERT, and its variants
- It's helpful on Natural Language Understanding tasks
  - Sequence Labeling & Semantic Matching

https://lemaoliu.github.io/retrieval-generation-tutorial/assets/slides/retrieval4lm.pdf

BERT and his muppet family (2018-)



GPT (2018)

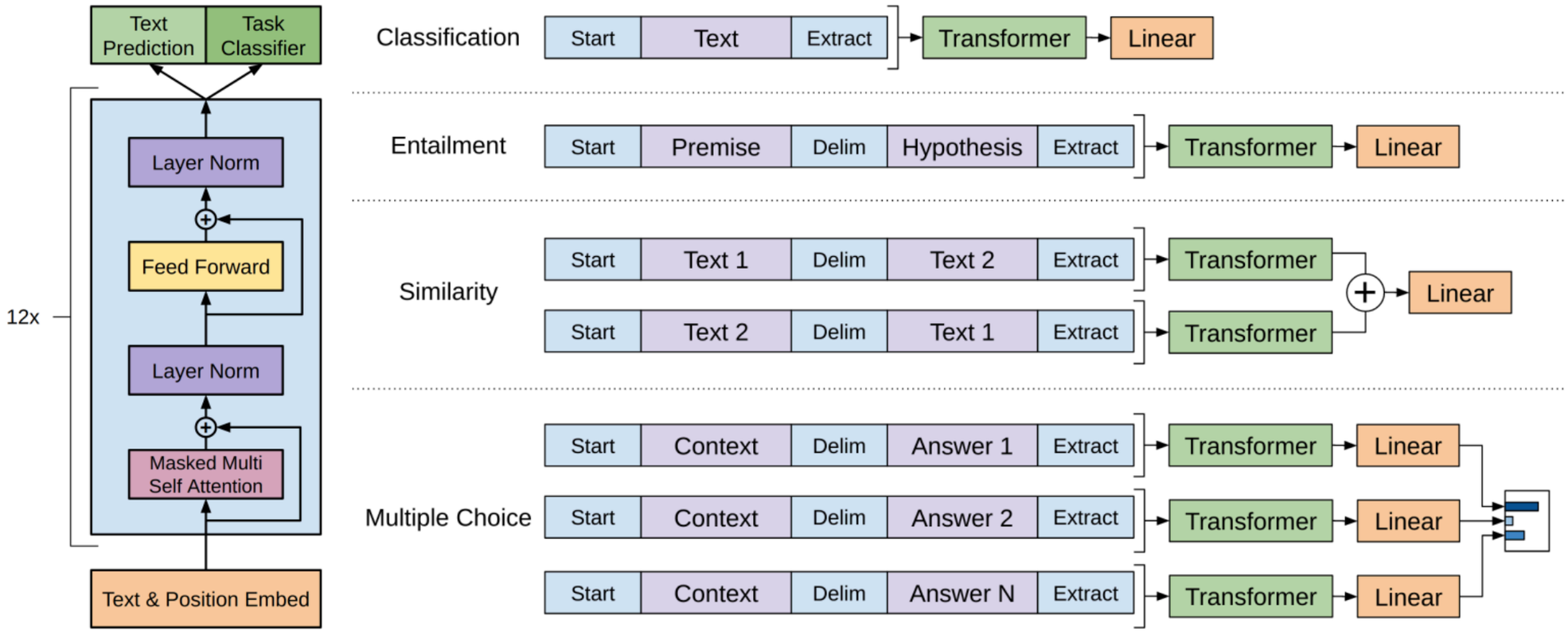


Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

GPT-2 (2019)

Language Models are Unsupervised Multitask Learners

Alec Radford [*1] **Jeffrey Wu** [*1] **Rewon Child** [1] **David Luan** [1] **Dario Amodei** [**1] **Ilya Sutskever** [**1]

Abstract

Natural language processing tasks, such as question answering, machine translation, reading comprehension, and summarization, are typically approached with supervised learning on task-specific datasets. We demonstrate that language models begin to learn these tasks without any explicit supervision when trained on a new dataset

competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent

“Emergent” abilities

GPT-2 uses the same architecture as GPT, just bigger (117M parameters —> 1.5B)

- But trained on much more data: 4GB -> 40GB of internet text data (WebText)
- Scrape links posted on Reddit w/ at least 3 upvotes (rough proxy of human quality)

One key emergent ability in GPT-2 is **zero-shot learning**: the ability to do many tasks with no examples, and no gradient updates

Zero-shot learning

For example, imagine we train a model on pictures of animals + text descriptions

The training data includes horses but no zebras

That is, it has no examples (or “shots”) of zebras

At “test” time, a model that can identify a zebra from a picture would be an example of zero-shot learning

Question	Generated Answer	Correct	Probability
Who wrote the book the origin of species?	Charles Darwin	✓	83.4%
Who is the founder of the ubuntu project?	Mark Shuttleworth	✓	82.0%
Who is the quarterback for the green bay packers?	Aaron Rodgers	✓	81.1%
Panda is a national animal of which country?	China	✓	76.8%
Who came up with the theory of relativity?	Albert Einstein	✓	76.4%
When was the first star wars film released?	1977	✓	71.4%
What is the most common blood type in sweden?	A	✗	70.6%
Who is regarded as the founder of psychoanalysis?	Sigmund Freud	✓	69.3%
Who took the first steps on the moon in 1969?	Neil Armstrong	✓	66.8%
Who is the largest supermarket chain in the uk?	Tesco	✓	65.3%
What is the meaning of shalom in english?	peace	✓	64.0%
Who was the author of the art of war?	Sun Tzu	✓	59.6%
Largest state in the us by land mass?	California	✗	59.2%
Green algae is an example of which type of reproduction?	parthenogenesis	✗	56.5%
Vikram samvat calender is official in which country?	India	✓	55.6%
Who is mostly responsible for writing the declaration of independence?	Thomas Jefferson	✓	53.3%
What us state forms the western boundary of montana?	Montana	✗	52.3%
Who plays ser davos in game of thrones?	Peter Dinklage	✗	52.1%
Who appoints the chair of the federal reserve system?	Janet Yellen	✗	51.5%
State the process that divides one nucleus into two genetically identical nuclei?	mitosis	✓	50.7%
Who won the most mvp awards in the nba?	Michael Jordan	✗	50.2%
What river is associated with the city of rome?	the Tiber	✓	48.6%
Who is the first president to be impeached?	Andrew Johnson	✓	48.3%
Who is the head of the department of homeland security 2017?	John Kelly	✓	47.0%
What is the name given to the common currency to the european union?	Euro	✓	46.8%
What was the emperor name in star wars?	Palpatine	✓	46.5%
Do you have to have a gun permit to shoot at a range?	No	✓	46.4%
Who proposed evolution in 1859 as the basis of biological development?	Charles Darwin	✓	45.7%
Nuclear power plant that blew up in russia?	Chernobyl	✓	45.7%
Who played john connor in the original terminator?	Arnold Schwarzenegger	✗	45.2%

Table 5. The 30 most confident answers generated by GPT-2 on the development set of Natural Questions sorted by their probability according to GPT-2. None of these questions appear in WebText according to the procedure described in Section 4.

GPT-3 (2020)

Language Models are Few-Shot Learners

Tom B. Brown* **Benjamin Mann*** **Nick Ryder*** **Melanie Subbiah***

Jared Kaplan† **Prafulla Dhariwal** **Arvind Neelakantan** **Pranav Shyam** **Girish Sastry**

Amanda Askell **Sandhini Agarwal** **Ariel Herbert-Voss** **Gretchen Krueger** **Tom Henighan**

Rewon Child **Aditya Ramesh** **Daniel M. Ziegler** **Jeffrey Wu** **Clemens Winter**

Christopher Hesse **Mark Chen** **Eric Sigler** **Mateusz Litwin** **Scott Gray**

Benjamin Chess **Jack Clark** **Christopher Berner**

Sam McCandlish **Alec Radford** **Ilya Sutskever** **Dario Amodei**

OpenAI

More “Emergent” abilities

GPT-3

- Another increase in size (1.5B parameters —> 175B)
- and training data (40GB —> over 600GB)

Ability to do emergent **few-shot learning**

- Specify a task by simply prepending examples of the task before your example
- Also called **in-context learning**, to stress that no gradient updates are performed when learning a new task

Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  cheese =>                           ← prompt
```

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

```
1  Translate English to French:
2  sea otter => loutre de mer
3  cheese =>
```

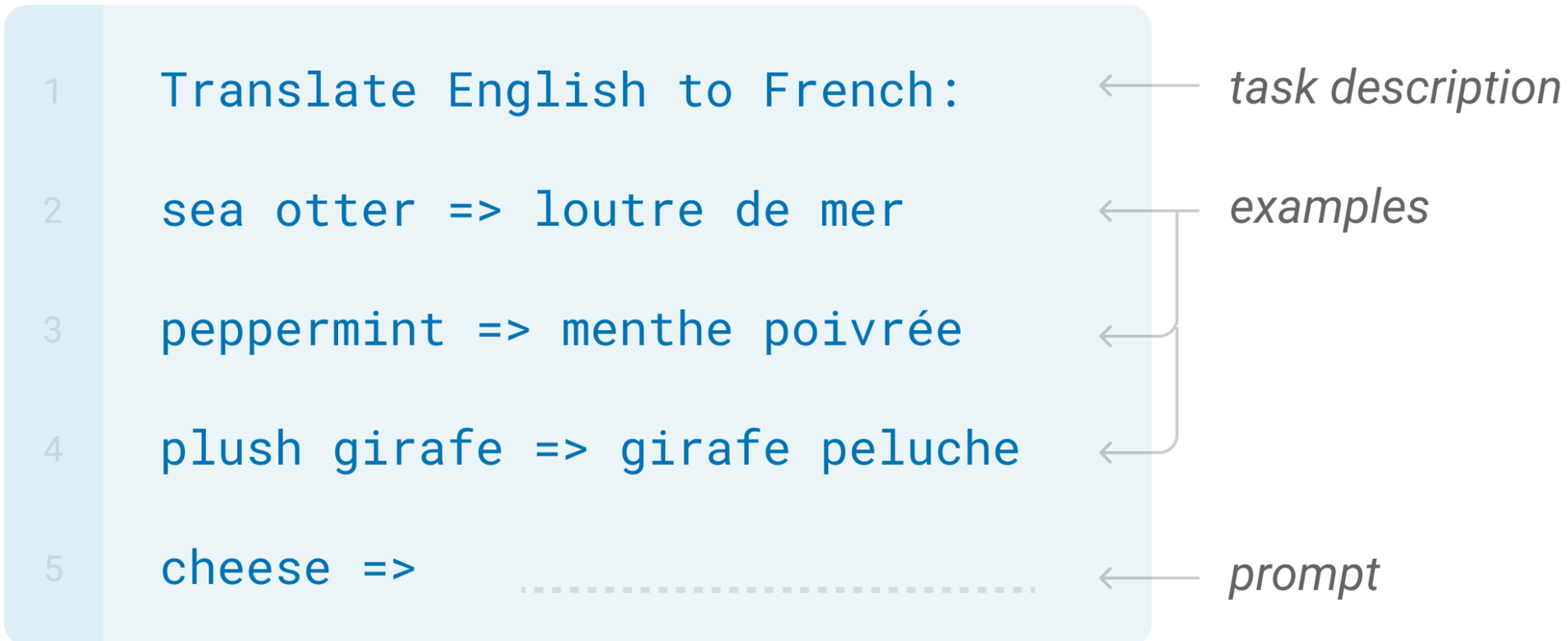
task description

example

prompt

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



Prompting

Send prompts to the LLM

- No examples: zero-shot
- One example: one-shot
- Several examples: few-shot

Activity 3

Part 0: Pair up (with a new partner!)

- Introduce yourself to your partner
- Share an interesting fact

Part 1 (Use Gemini 1.5 Flash)

- Use either the browser or colab access to Gemini Flash 1.5
- Identify a zero-shot, one-shot, few-shot prompting example
  - E.g., could be labeling, translation, jokes, math problems, ...

Activity 3

Activity 3

Activity 3