

A Random Walk Down GPT Street

Presented by
David Greenwood

a light hearted exploration
of the ever accelerating
LLM landscape.

Overview

04 Why the hype?

09 What is a GPT? Why now?

12 How it works?

18 How to give it a go?

24 GPT on laptop?

26 Where to find out more?



Hello!

I'm Dr David Greenwood, a Data Scientist & Systems Engineer based out of the green pastures of Wiltshire. I specialize in applying Data Science to urban regeneration, investment management & risk management use cases.

GPT? Why the hype?

"Why the hype?"

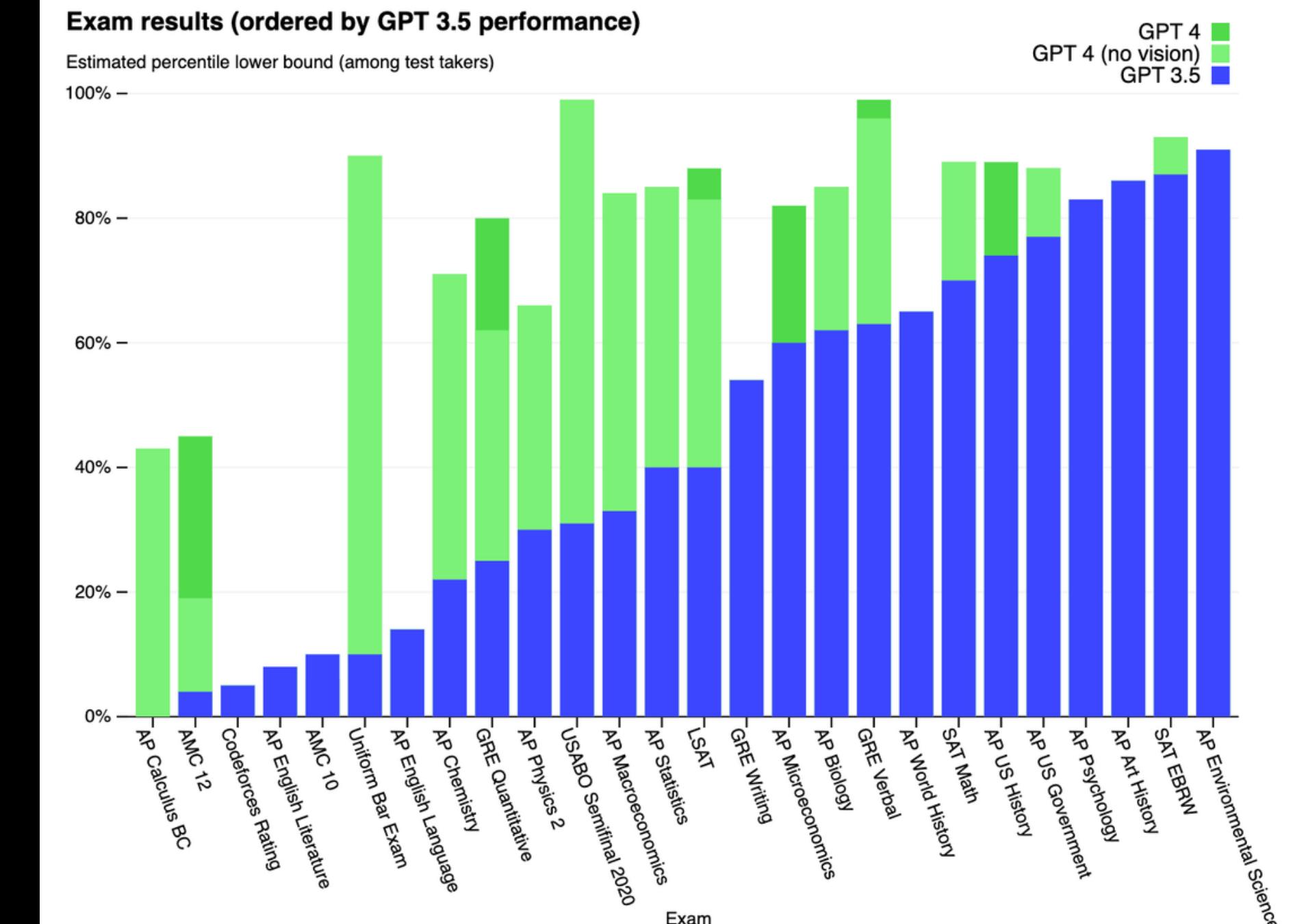
A polymath that augments your team
and never gets tired.

How many people work for you?

How many have +80th percentile scores
in Math, Verbal, Stats, Macroeconomics,
Biology and the Bar exam?

How many are willing to work for almost
free & never get tired?

GPT-4 is a polymath, waiting for you to
ask for help.



"Why the hype?"

The democratisation of Data Science

What subject does the following text cover? Choose the best category or topic.

Alexander Boris de Pfeffel Johnson (/'fɛfəl/, [5] born 19 June 1964) is a British politician, writer and journalist who served as Prime Minister of the United Kingdom and Leader of the Conservative Party from 2019 to 2022. He previously served as Foreign Secretary from 2016 to 2018 and as Mayor of London from 2008 to 2016. Johnson has been Member of Parliament (MP) for Uxbridge and South Ruislip since 2015, having previously been MP for Henley from 2001 to 2008.

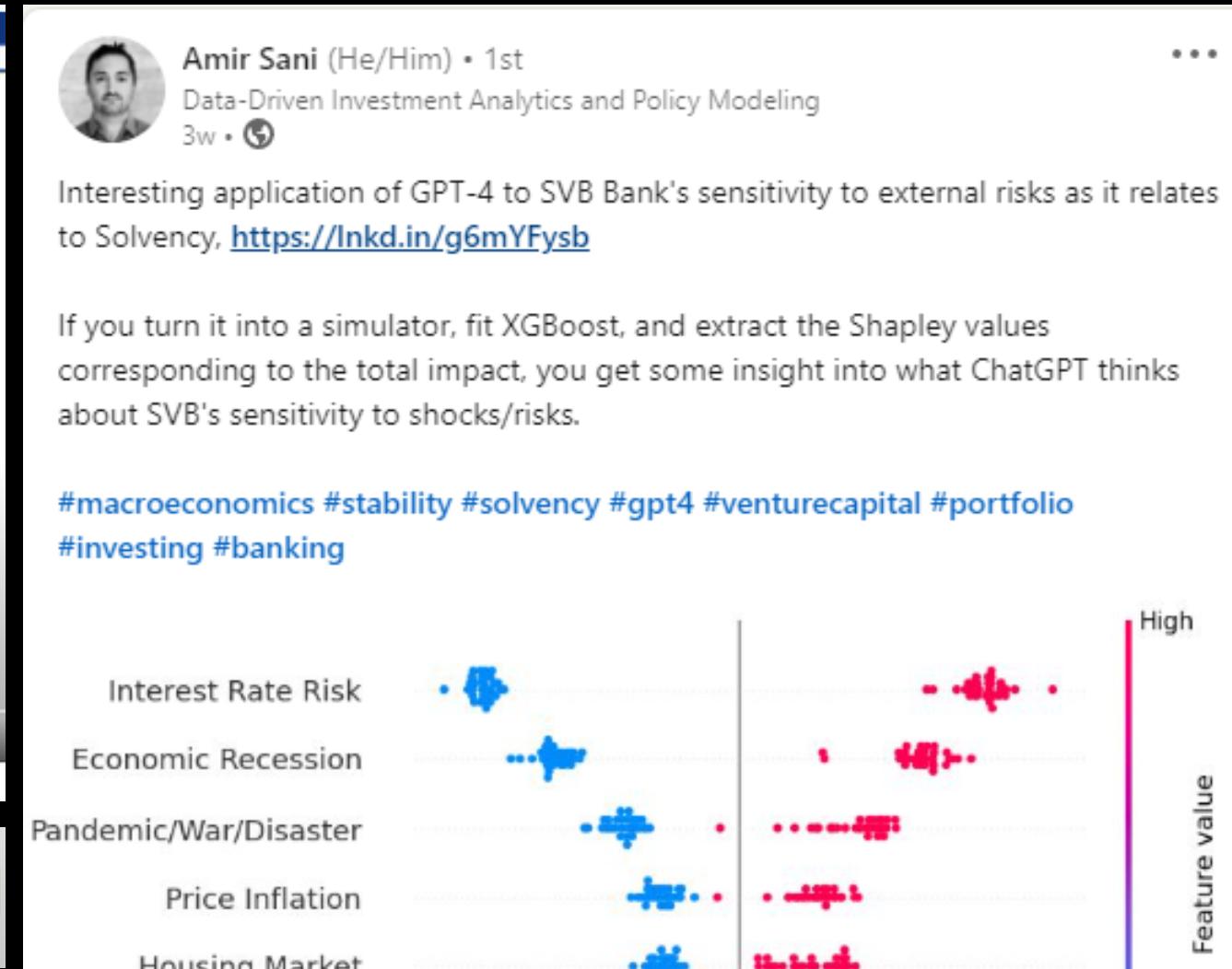
Johnson attended Eton College, and studied Classics at Balliol College, Oxford. He was elected President of the Oxford Union in 1986. In 1989, he became the Brussels correspondent – and later political columnist – for The Daily Telegraph, and from 1999 to 2005 he was the editor of The Spectator. Following his election to Parliament in 2001, he became a member of the shadow cabinets of Michael Howard and David Cameron. In 2008, Johnson was elected Mayor of London and resigned from the House of Commons. He was re-elected mayor in 2012. At the 2015 general election he was elected MP for Uxbridge and South Ruislip, and the following year did not seek re-election as mayor. Johnson was a prominent figure in the successful Vote Leave campaign for Brexit in the 2016 European Union membership referendum. After the referendum, Theresa May appointed him as Foreign Secretary in her cabinet. He resigned from the position two years later in protest against the Chequers Agreement and May's approach to Brexit.

British politics and the career of Alexander Boris de Pfeffel Johnson.

The screenshot shows a data analysis application interface. At the top, there is a navigation bar with a logo, a search bar, and several menu items. Below the navigation bar, there is a section titled "Database views" which contains a table with employee data. The table has columns for employee_id, full_name, job_title, department, business_unit, gender, ethnicity, age, and hi. There are three rows of data: 0 E02002 Kai Le Controls Engineer Manufacturing Male Asian 47 2C 0C, 1 E02003 Robert Patel Analyst Sales Corporate Male Asian 58 2C 0C, and 2 E02004 Cameron Lo Network Administrator IT Research & Development Male Asian 34 2C 0C. Below the database view, there is a "Query Data" section with a text input field containing the question "Who is paid the most?". The response "Robert Rogers is paid the most." is displayed below the input field.

Classification

Data Analysis



Risk Analysis

"Why the hype?"

Democratisation of knowledge work

```
j = capsys.readouterr() assert "Invalid" to basic_math.py run the tests and check for correct func

: {'file': 'basic_math.py', 'text': '\narea(0), 0, rel_tol=1e-9)\n assert math.isclose(calc_area(2), 4 * math.pi, rel_tol=1e-9)\n assert math.isclose(calc_circumference(1), 2 * math.pi, rel_tol=1e-9)\n assert math.isclose(calc_circumference(2), 4 * math.pi, rel_tol=1e-9)\n\nArea of the circle: 18.84955592153876\n' in captured.outerr()\n assert "Invalid input. Please ended successfully.\nmath.py file to test the code.\nworks as expected and the functions are'

ENTS = {'file': 'basic_math.py'}
line 6 def calc_circumference()
fixing the syntax issue.
fix it to ensure the code works as intended.
```

AutoGPT

<https://twitter.com/i/status/164218>

1498278408193

```
= capsys.readouterr() assert "Invalid" to basic_math.py run the tests and check for correct func

{'file': 'basic_math.py', 'text': '\narea(0), 0, rel_tol=1e-9)\n assert math.isclose(calc_area(2), 4 * math.pi, rel_tol=1e-9)\n assert math.isclose(calc_circumference(1), 2 * math.pi, rel_tol=1e-9)\n assert math.isclose(calc_circumference(2), 4 * math.pi, rel_tol=1e-9)\n\nArea of the circle: 18.84955592153876\n' in captured.outerr()\n assert "Invalid input. Please ended successfully.\nmath.py file to test the code.\nworks as expected and the functions are'

ENTS = {'file': 'basic_math.py'}
line 6 def calc_circumference()
fixing the syntax issue.
fix it to ensure the code works as intended.
```

"Any sufficiently advanced technology is indistinguishable from magic."

Arthur C. Clarke

What is a GPT, Why now?

What is GPT?

GPT

Generative

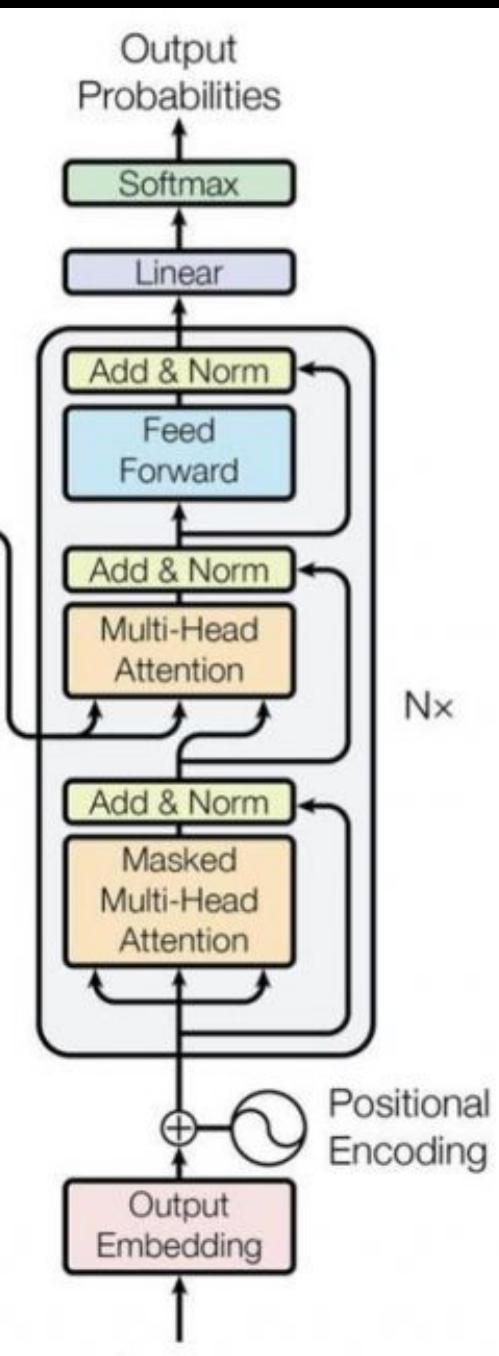
Produces content (from "joint probabilities")

Pre-trained

Comes out-of-the-box with "some ability"

Transformer

one type of input into another type of output



Why Now?

GPU's

+

novel Transformer
architectures that are highly
parallelizable

=

Unique moment in history

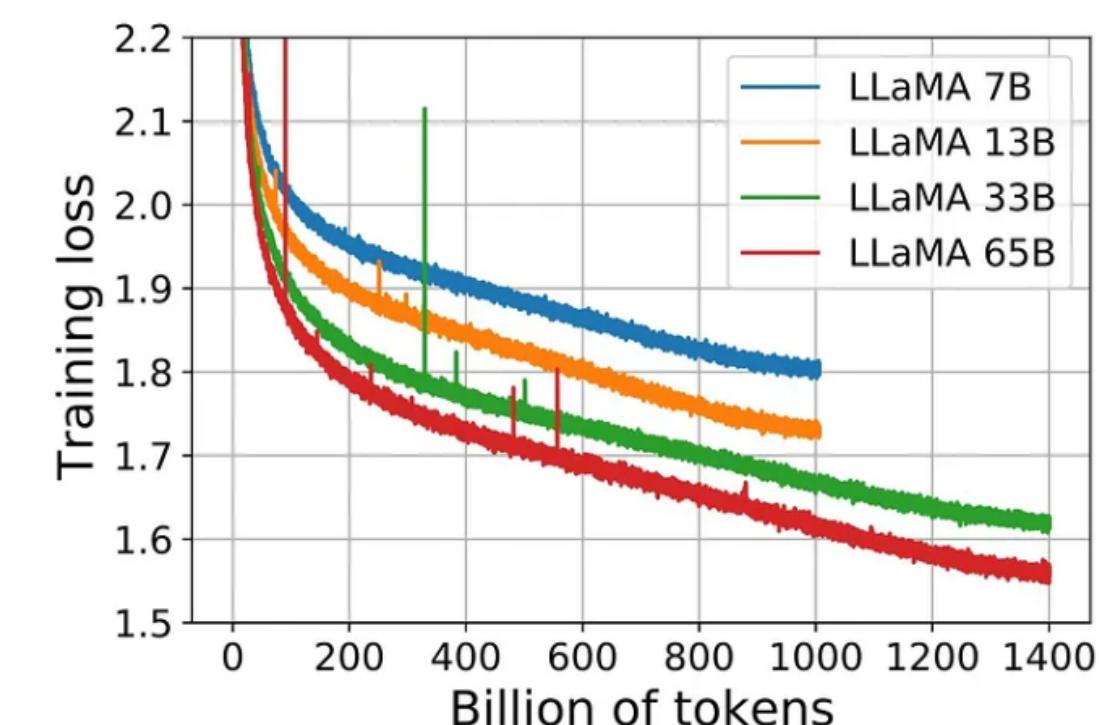
<https://towardsdatascience.com/behind-the-millions-estimating-the-scale-of-large-language-models-97bd7287fb6b>

Release	Model	Size	Paper
2019	GPT-2	1.5B	Language Models are Unsupervised Multitask Learners
2020	GPT-3	175B	Language Models are Few-Shot Learners
2021	Gopher	280B	Scaling Language Models: Methods, Analysis & Insights from Training Gopher
2022	PaLM	540B	PaLM: Scaling Language Modeling with Pathways
2022	Chinchilla	70B	Training Compute-Optimal Large Language Models
2022	OPT	175B	OPT: Open Pre-trained Transformer Language Models
2022	BLOOM	176B	BLOOM: A 176B-Parameter Open-Access Multilingual Language Model
2022	Galactica	120B	Galactica: A Large Language Model for Science
2023	LLaMA	65B	LLaMA: Open and Efficient Foundation Language Models

Some of the popular LLMs architectures. Image by Author

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

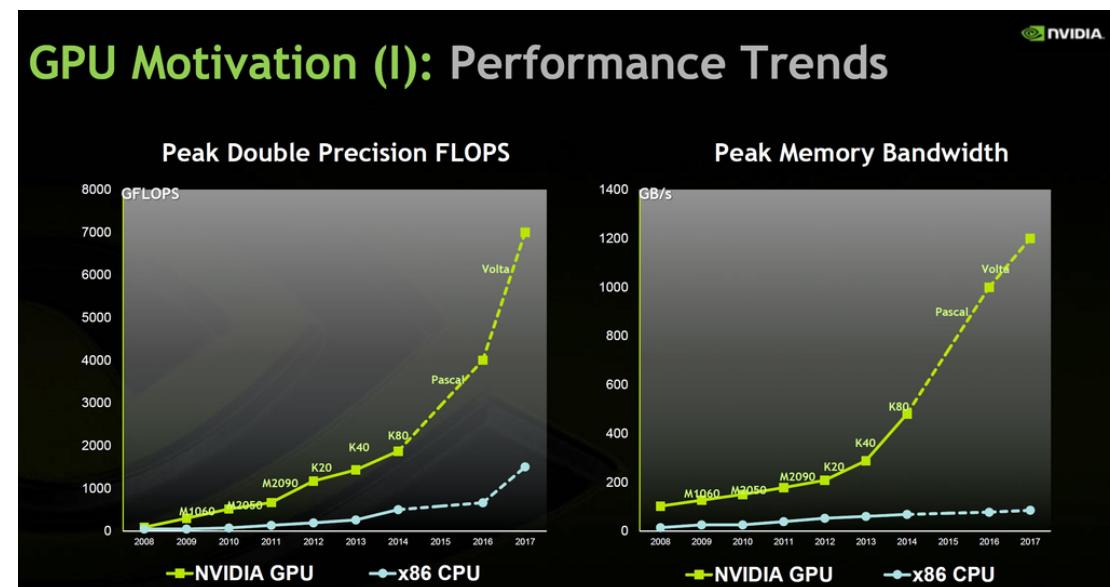
Table 1: **Pre-training data.** Data mixtures used for pre-training, for each subset we list the sampling proportion, number of epochs performed on the subset when training on 1.4T tokens, and disk size. The pre-training runs on 1T tokens have the same sampling proportion.



When training a 65B-parameter model, our code processes around 380 tokens/sec/GPU on 2048 A100 GPU with 80GB of RAM. This means that training over our dataset containing 1.4T tokens takes approximately 21 days.

2048 GPUs x \$3.93 GPU per hour x 24 hours x 21 days =

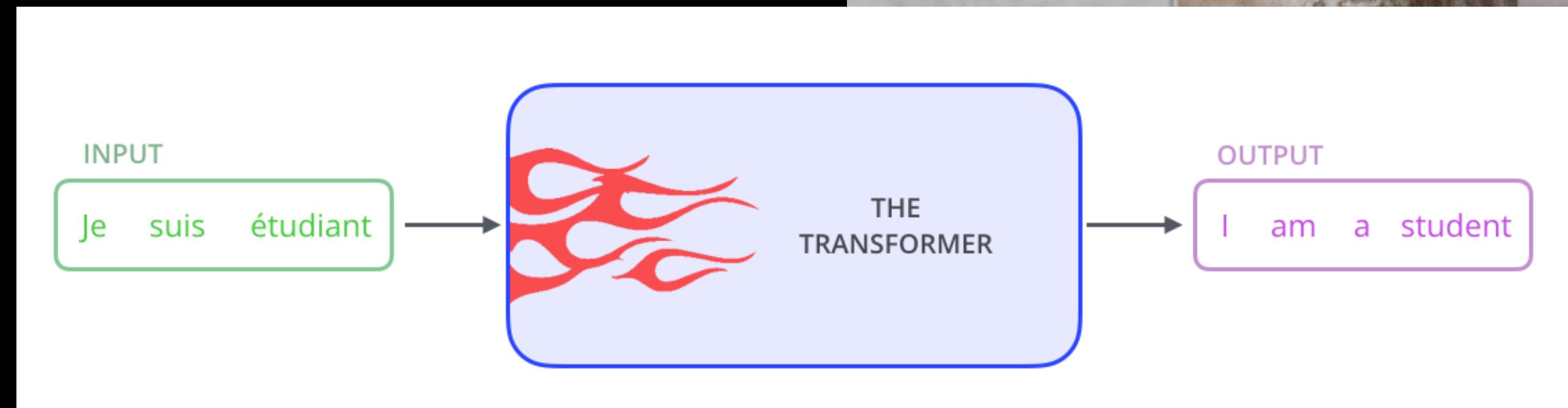
4.05 million dollars



What is the
Transformer
architecture?
How does it work?

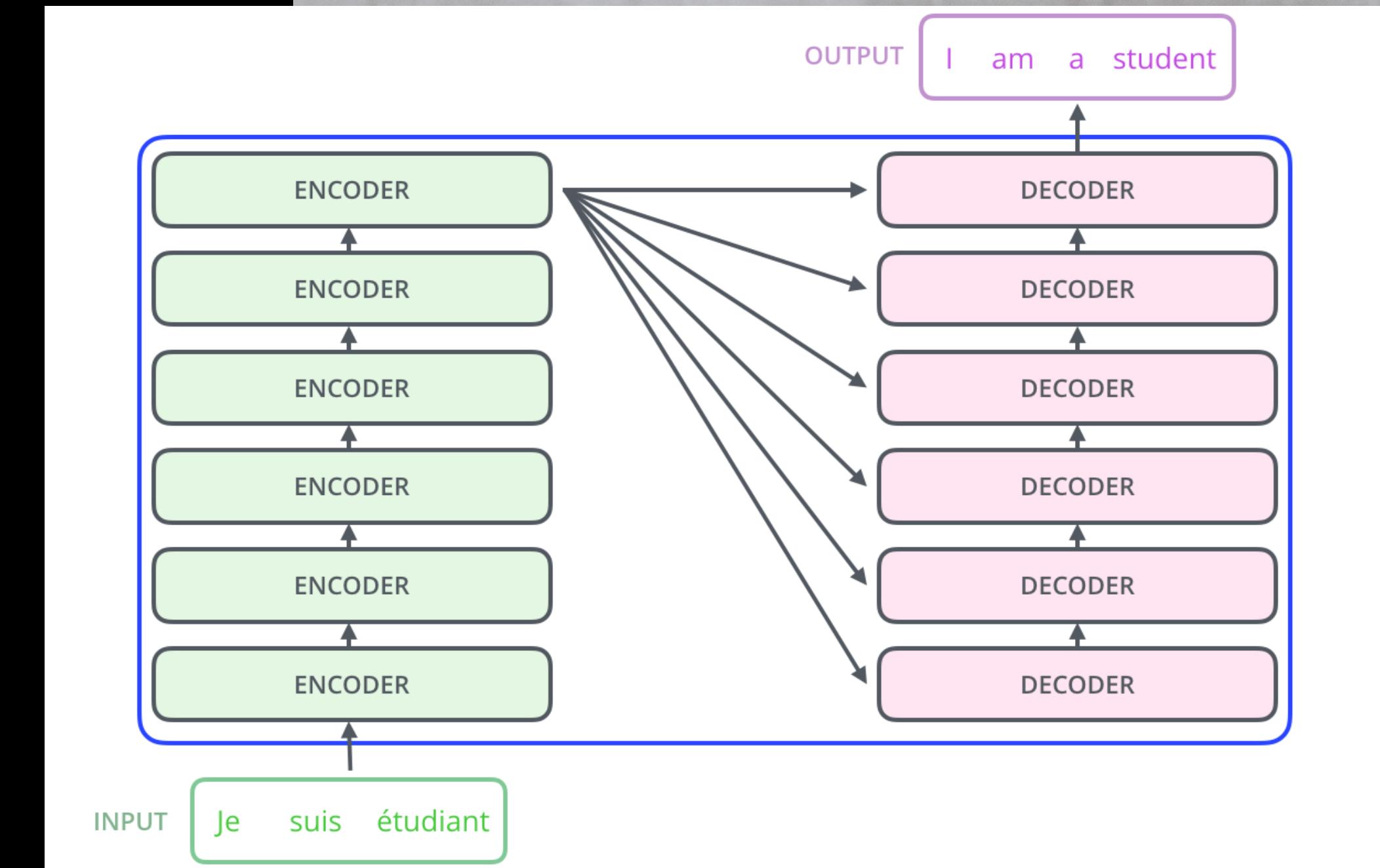
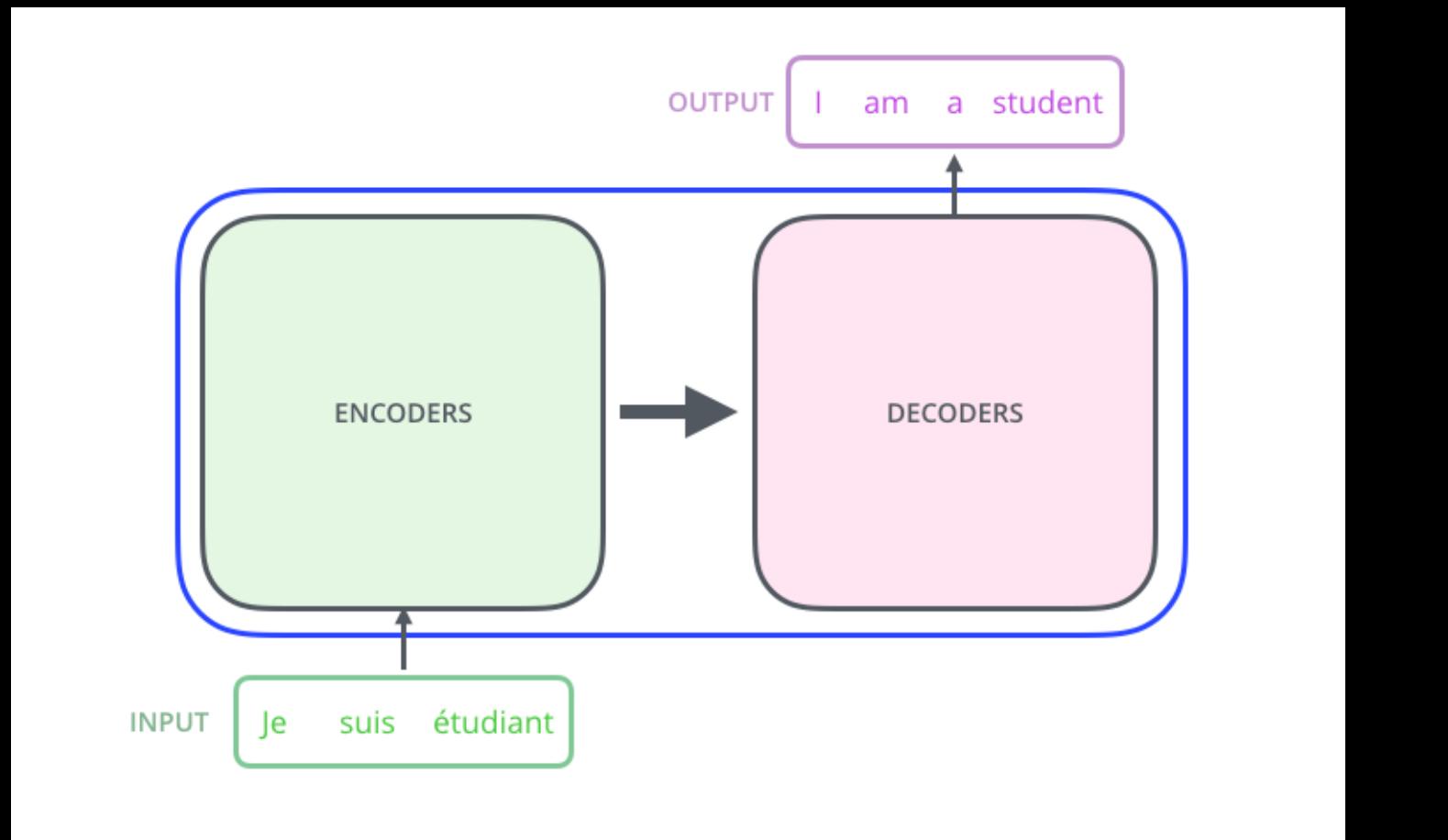
The Transformer

Input - Output

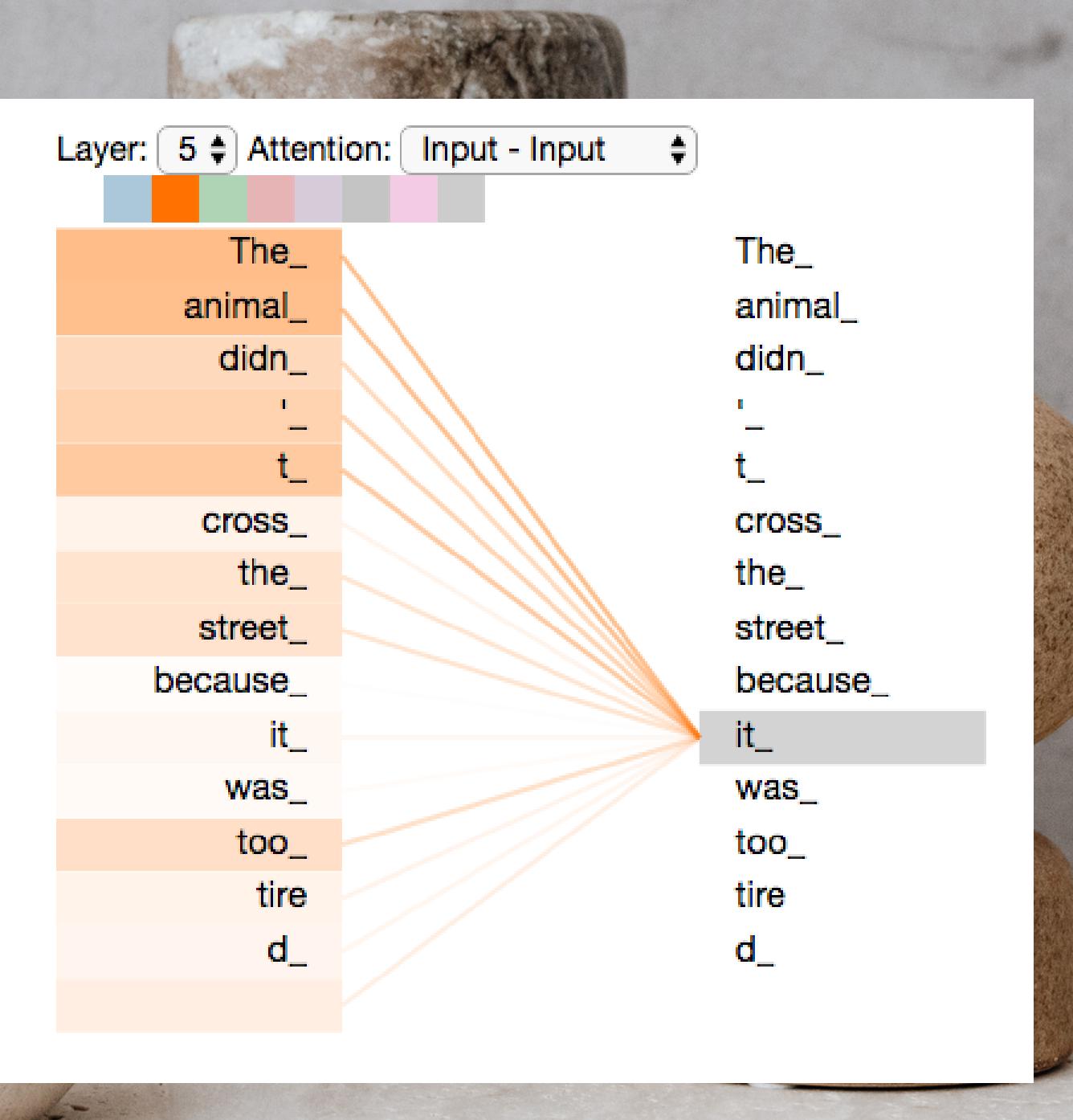
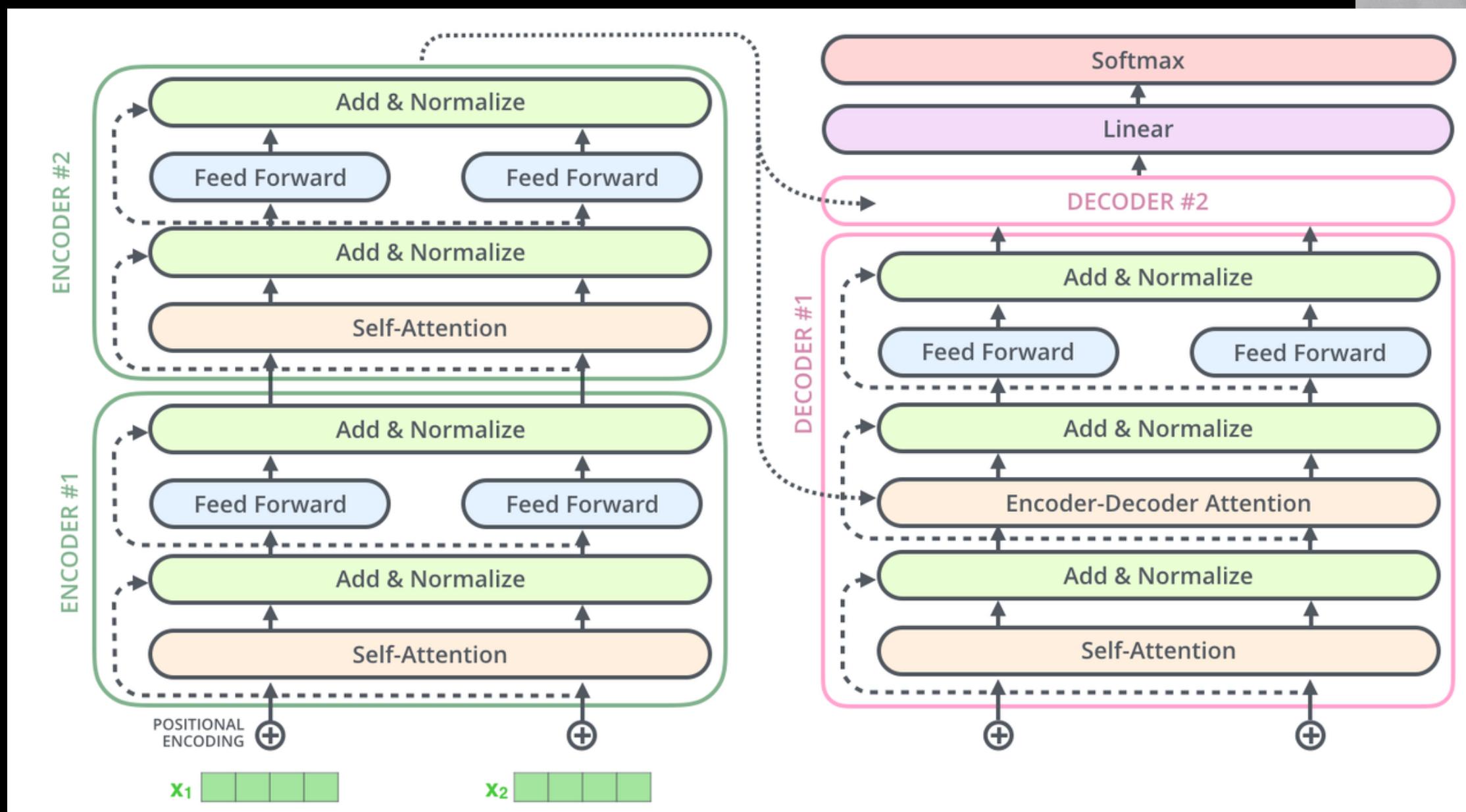


The Transformer

Basic Architecture



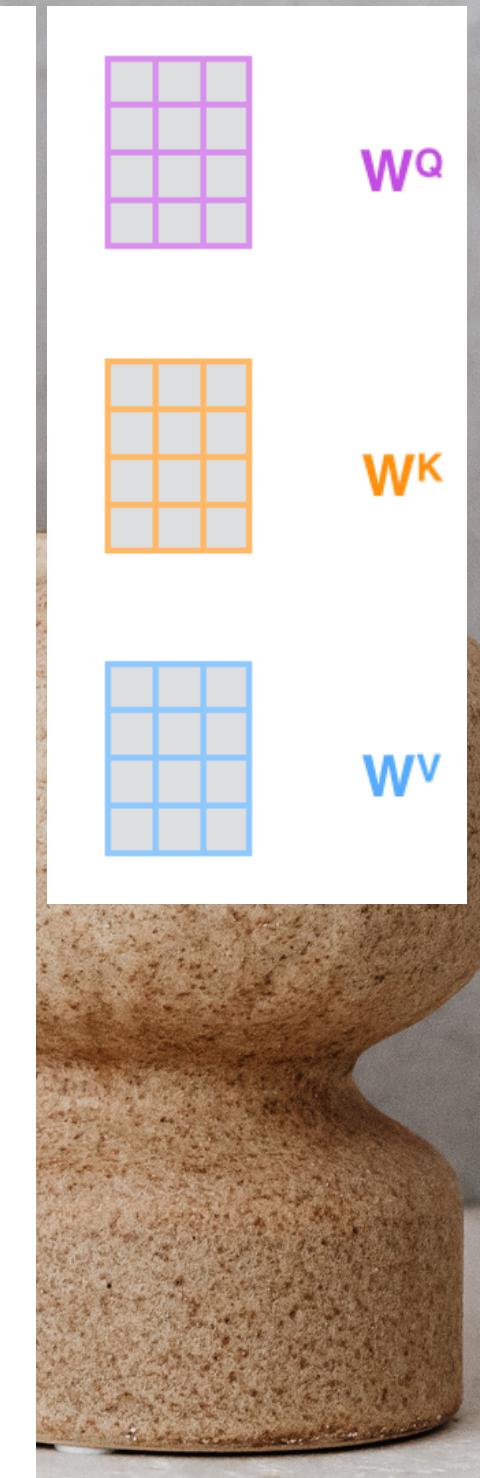
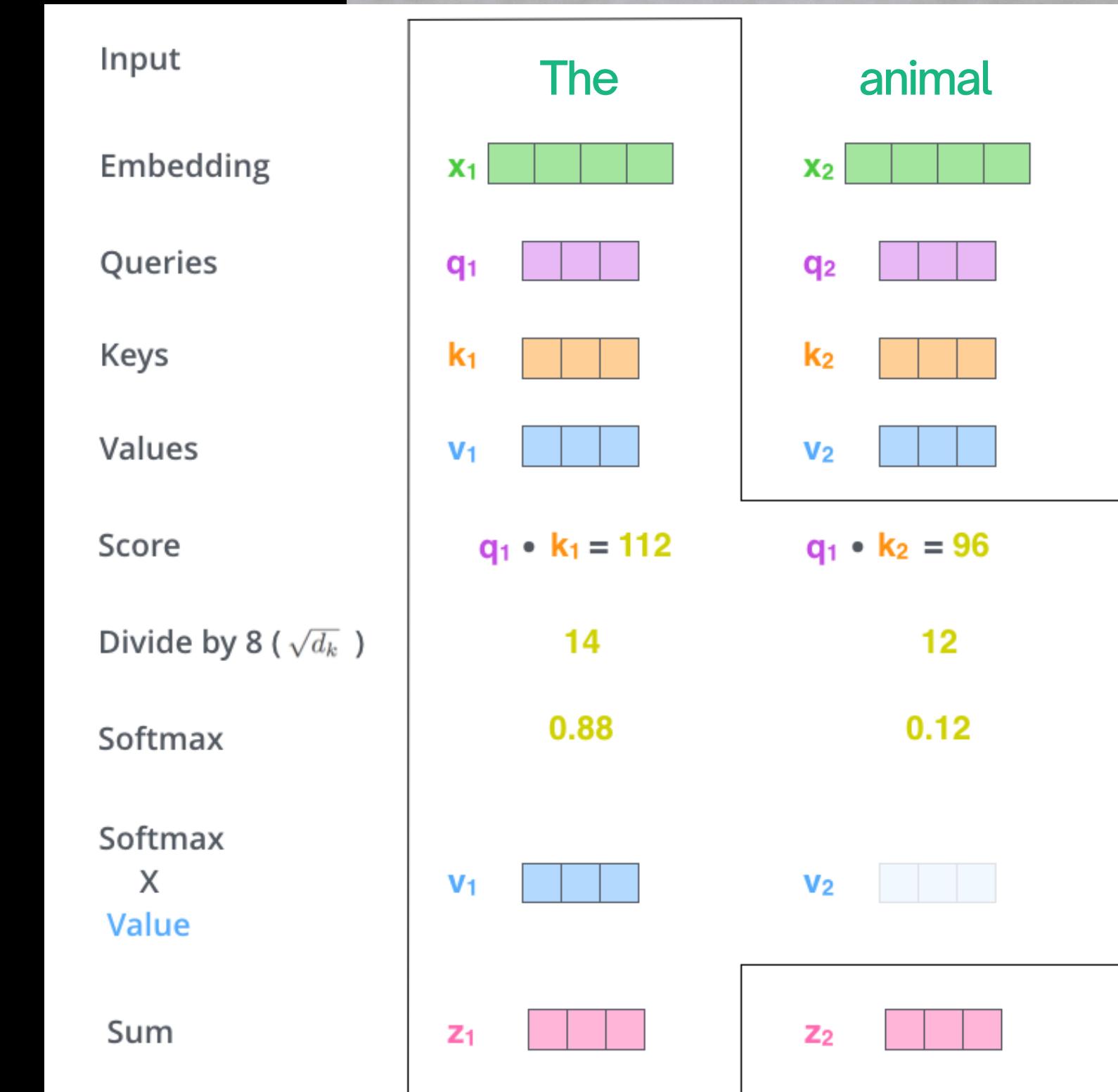
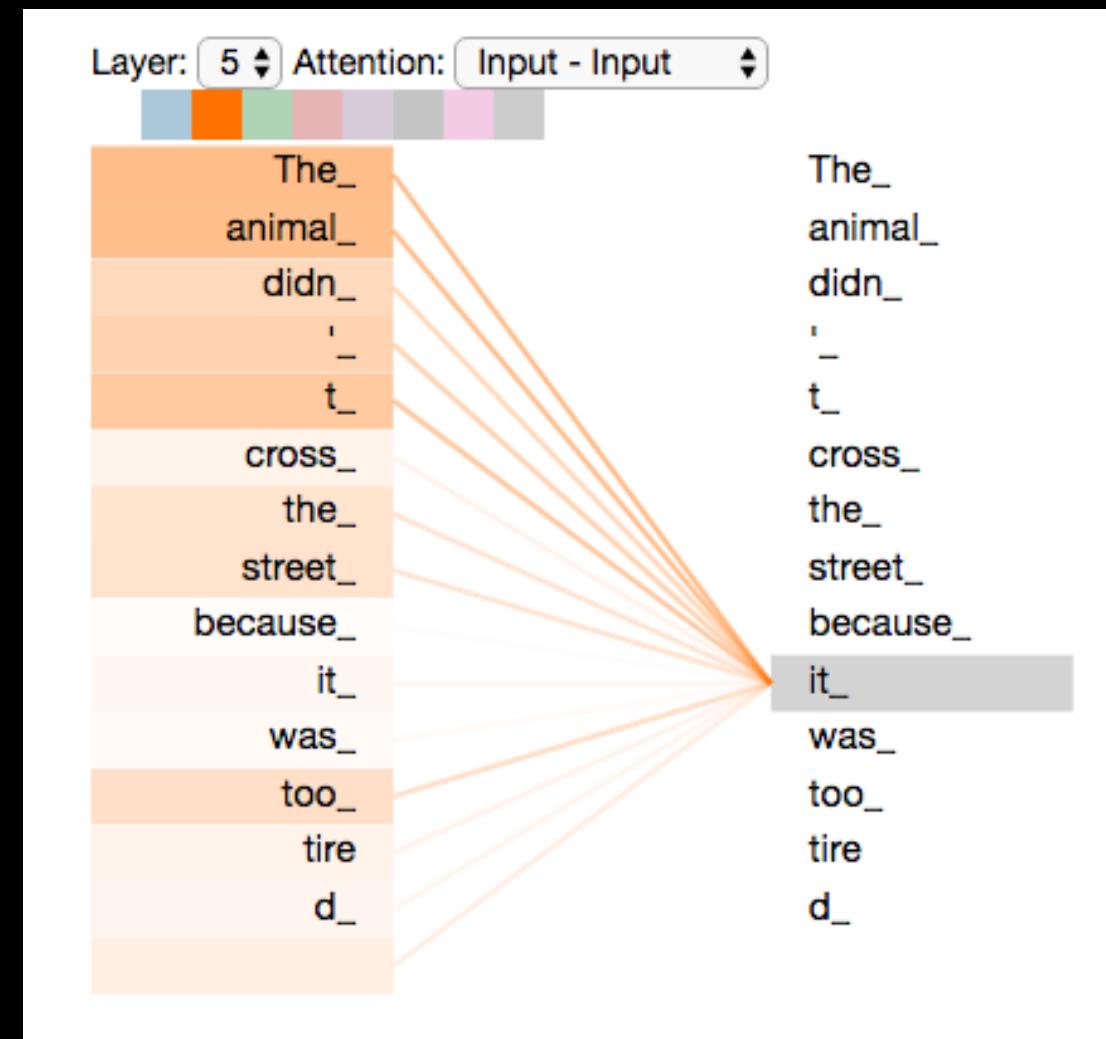
The Transformer Architecture



"The animal didn't cross the street because it was too tired"

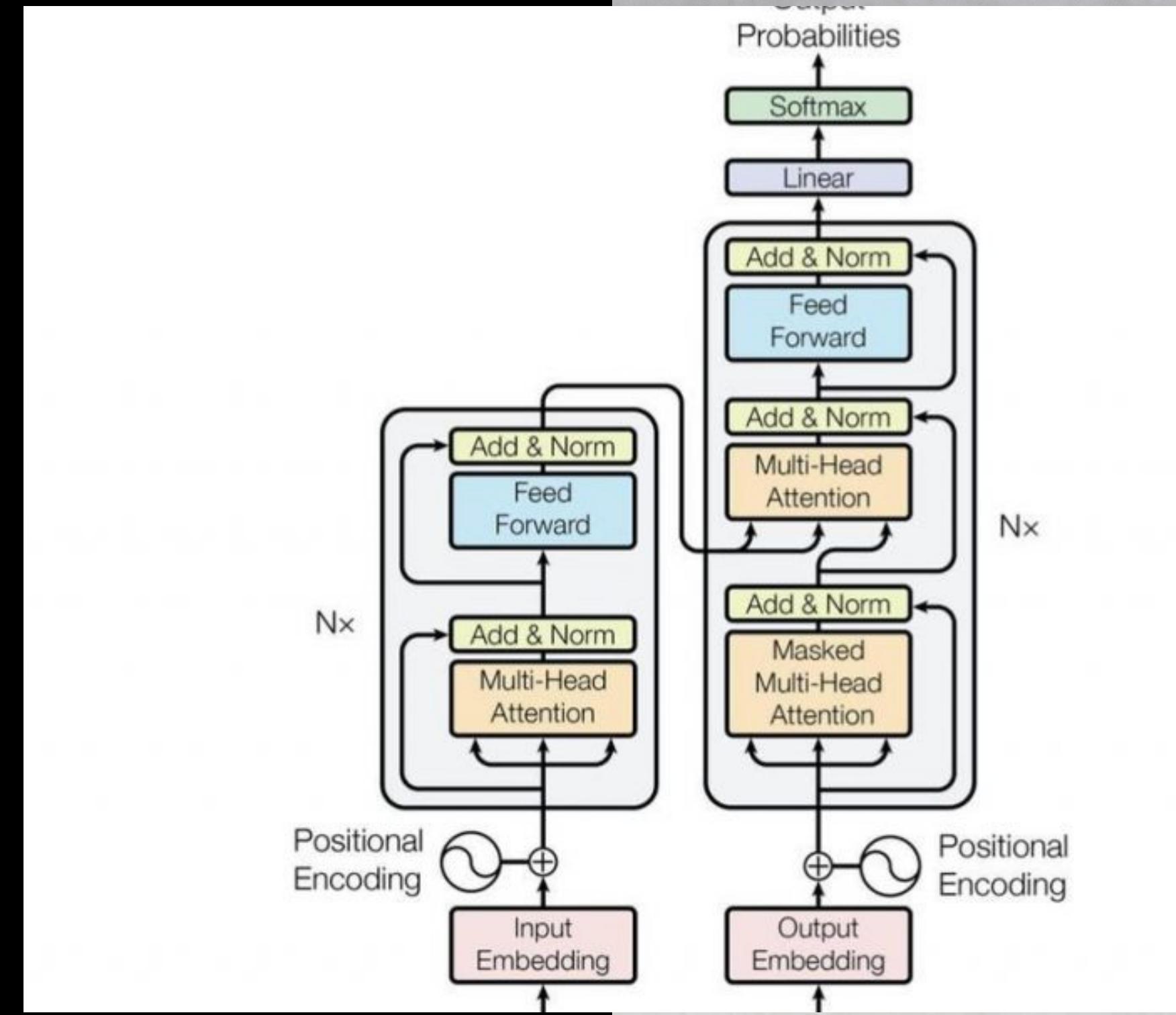
The Transformer

Self Attention



The Transformer

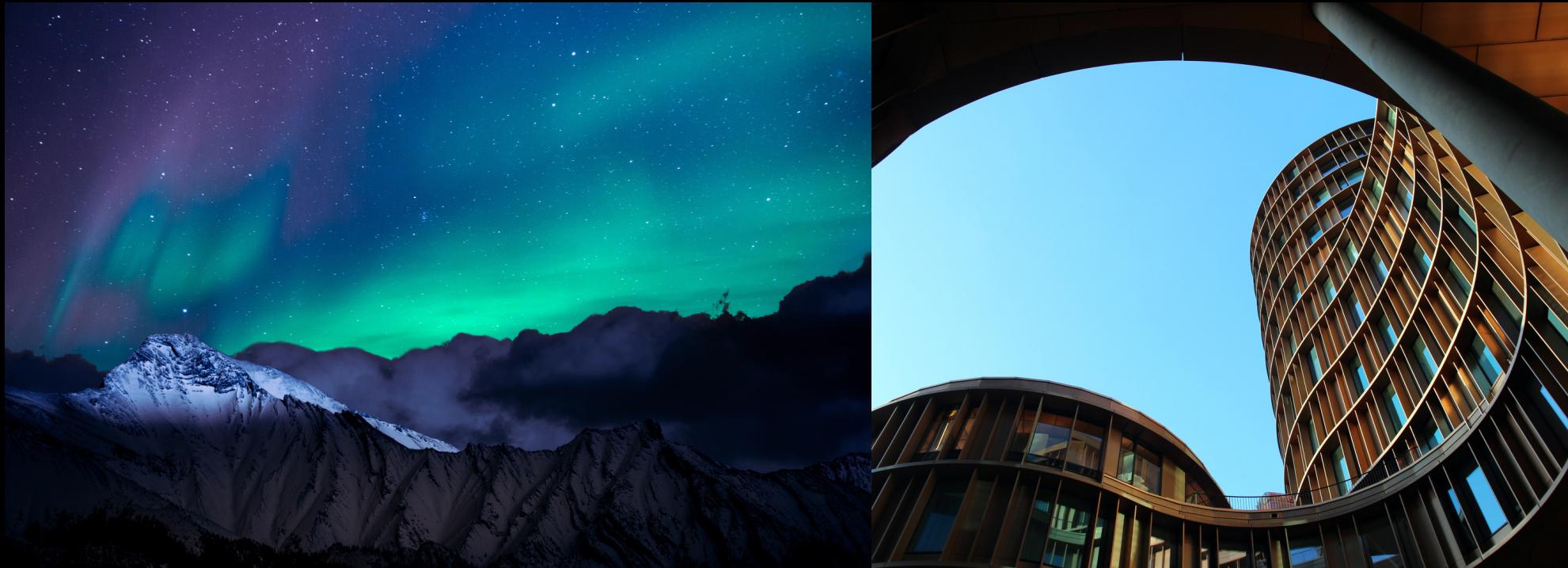
All Together



How do I give it a go?

Choices?

Different ways of "conditioning" your model depending on number of examples



Prompt engineering

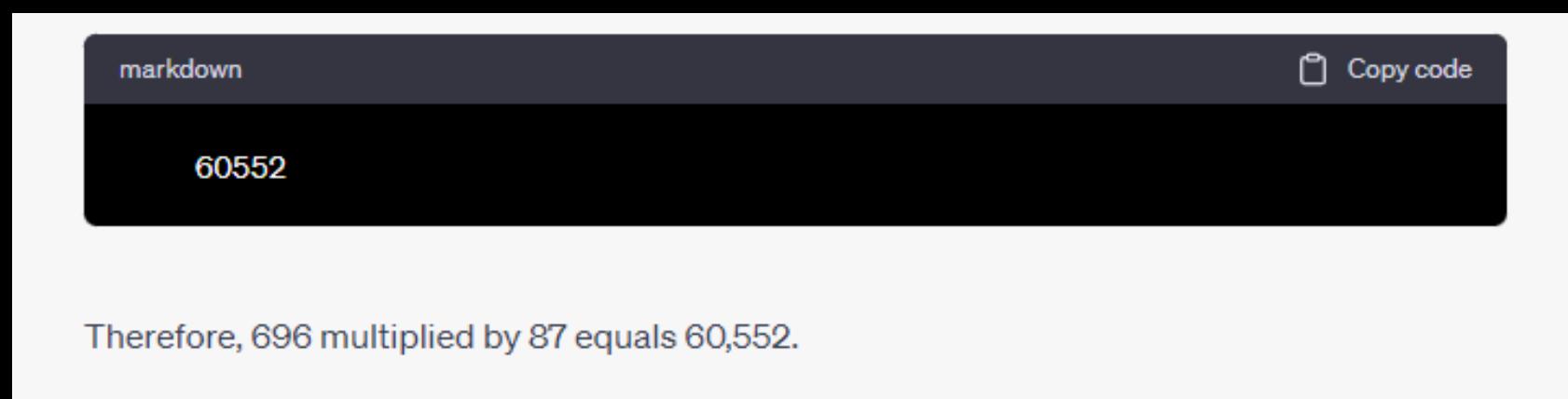
Fine tuning

```
1 {"prompt": "<prompt text>", "completion": "<ideal generated text>"}
2 {"prompt": "<prompt text>", "completion": "<ideal generated text>"}
```

Both are forms of model conditioning

Prompt Engineering Examples

Different ways of "conditioning" your model



5. Start with zero-shot, then few-shot (example), neither of them worked, then fine-tune

Zero-shot

Extract keywords from the below text.

Text: {text}

Keywords:

Few-shot - provide a couple of examples

Extract keywords from the corresponding texts below.

Text 1: Stripe provides APIs that web developers can use to integrate payment processing.

Keywords 1: Stripe, payment processing, APIs, web developers, websites, mobile applicat:

##

Text 2: OpenAI has trained cutting-edge language models that are very good at understand

Keywords 2: OpenAI, language models, text processing, API.

##

Text 3: {text}

Keywords 3:

Open-source LLMs

There is a whole Zoo of stuff out there!

		BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA
GPT-3	175B	60.5	81.0	-	78.9	70.2	68.8	51.4	57.6
Gopher	280B	79.3	81.8	50.6	79.2	70.1	-	-	-
Chinchilla	70B	83.7	81.8	51.3	80.8	74.9	-	-	-
PaLM	62B	84.8	80.5	-	79.7	77.0	75.2	52.5	50.4
PaLM-cont	62B	83.9	81.4	-	80.6	77.0	-	-	-
PaLM	540B	88.0	82.3	-	83.4	81.1	76.6	53.0	53.4
LLaMA	7B	76.5	79.8	48.9	76.1	70.1	72.8	47.6	57.2
	13B	78.1	80.1	50.4	79.2	73.0	74.8	52.7	56.4
	33B	83.1	82.3	50.4	82.8	76.0	80.0	57.8	58.6
	65B	85.3	82.8	52.3	84.2	77.0	78.9	56.0	60.2

Table 3: Zero-shot performance on Common Sense Reasoning tasks.

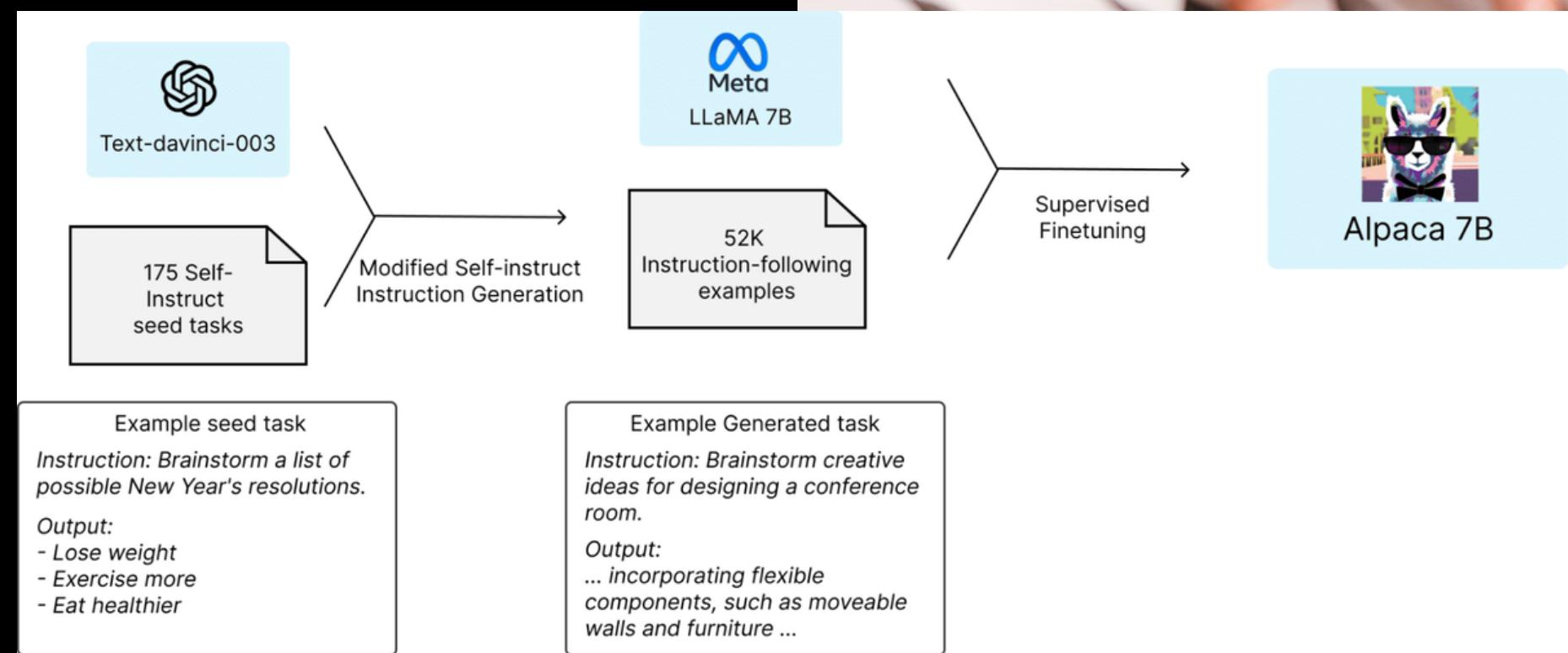
The Meta LLaMA release triggered a range of open source models from which to pick and choose.



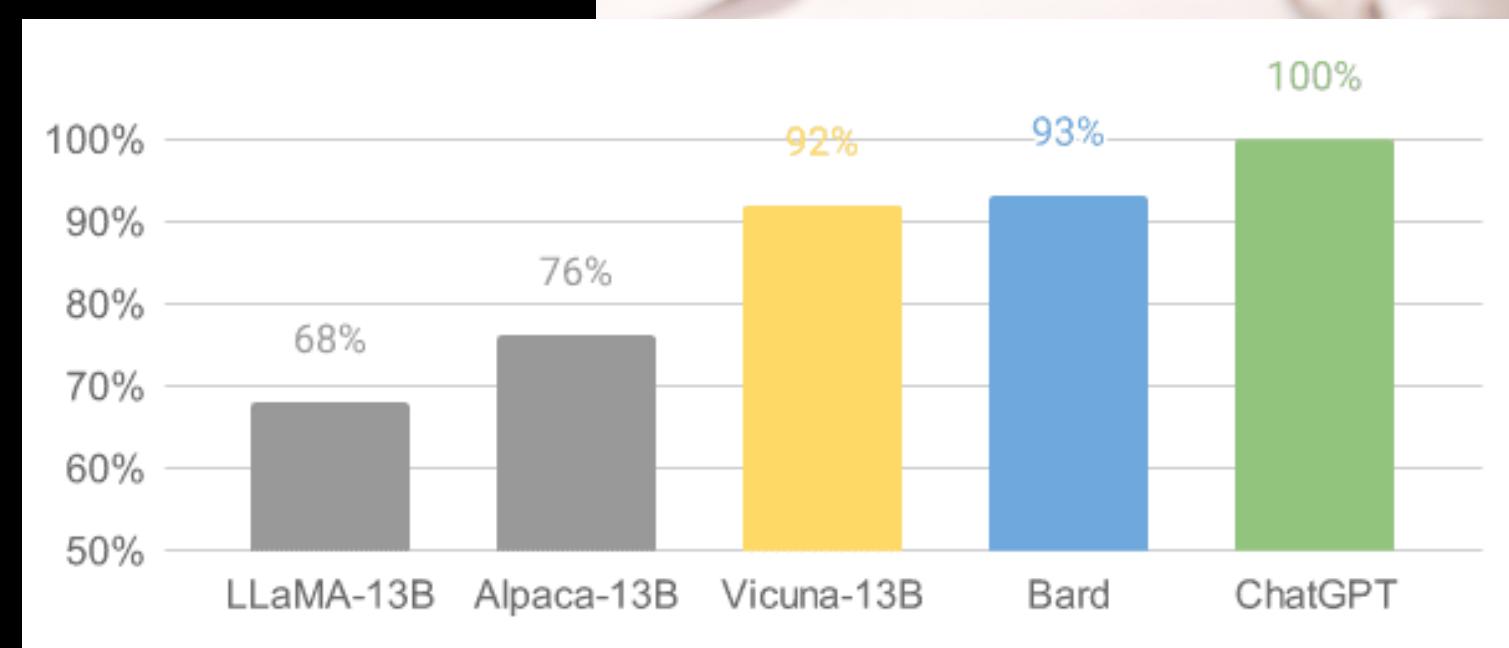
Open-source LLMs

There is a whole Zoo of stuff out there!

Alpaca



Vicuna



Open-source LLMs

A good place to get started?

The screenshot shows the Hugging Face NLP Course landing page. It features a large search bar at the top with the placeholder "Search models, datasets, users...". Below the search bar is a "NLP Course" section with a dropdown menu. To the right of the dropdown is a graduation cap emoji and a "Ctrl+K" keyboard shortcut. Further down are "Search documentation" and "MAIN EN" buttons, along with a "1,000" badge. A sidebar on the right contains links to "Introduction", "Transformer models", "Using 😊 Transformers", "Fine-tuning a pretrained model", and "Sharing models and tokenizers".

The screenshot shows the "Introduction" page of the Hugging Face NLP Course. It lists five main topics: "Transformer models", "Using 😊 Transformers", "Fine-tuning a pretrained model", and "Sharing models and tokenizers". Each topic has a corresponding button.

The screenshot shows the model card for "vicuna-13b-delta-v1.1" by lm-sys. The card includes details like "Text Generation", "PyTorch", "Transformers", "llama", and "License: apache-2.0". It also features buttons for "Train", "Deploy", and "Use in Transformers". The "Model card" tab is selected. A note states: "NOTE: This 'delta model' cannot be used directly. Users have to apply it on top of the original LLaMA weights to get actual Vicuna weights. See <https://github.com/lm-sys/FastChat#vicuna-weights> for instructions." A chart shows "Downloads last month" at 1,468. Other sections include "Hosted inference API" and "Text Generation".

The screenshot shows a Jupyter Notebook cell with the following code:

```
[23]: from transformers import pipeline  
  
question_answerer = pipeline("question-answering")  
question_answerer(  
    question="Who is a scientist?",  
    context="Sylvain is a database analyst. David is a data analyst. Brian is a physicist.",  
)
```

Output:

```
No model was supplied, defaulted to distilbert-base-cased-distilled-squad and revision 626af31 (https://huggi  
Using a pipeline without specifying a model name and revision in production is not recommended.  
{'score': 0.9582207202911377, 'start': 56, 'end': 61, 'answer': 'Brian'}
```

**HTTPS://HUGGINGFACE.CO/
LEARN/NLP-COURSE/**



GPT from the
comfort of your
laptop?

The mandatory live demo that goes tediously wrong, despite it previously working great

Resource Page

HuggingFace

<https://huggingface.co/learn/nlp-course/>

Illustrated Transformer

<http://jalammar.github.io/illustrated-transformer/>

How to train your own Large Language Models

<https://blog.replit.com/llm-training>

Understanding Large Language Models

<https://magazine.sebastianraschka.com/p/understanding-large-language-models>

LLaMA: Open and Efficient Foundation Language Models

<https://arxiv.org/abs/2302.13971>

Best practices for prompt engineering with OpenAI

<https://help.openai.com/en/articles/6654000-best-practices-for-prompt-engineering-with-openai-api>

to get in
touch.

Linked-in

<https://www.linkedin.com/in/dsgreenwood/>

Email

dsg27@cantab.net