CS 232: Artificial Intelligence

Fall 2023

Prof. Carolyn Anderson
Wellesley College

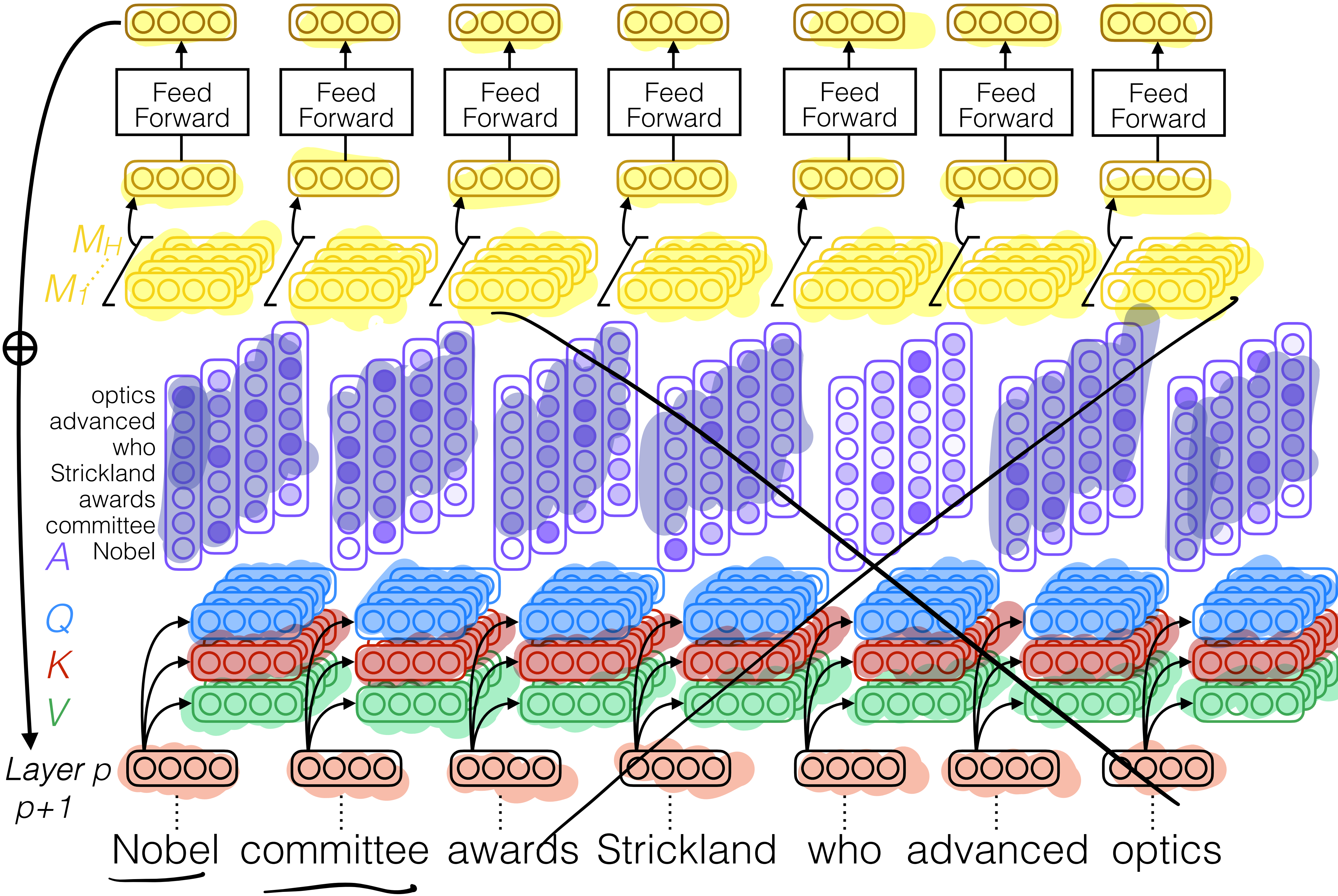
Recap

Transformers

Slides by Emma Strubell!

[Vaswani et al. 2017]

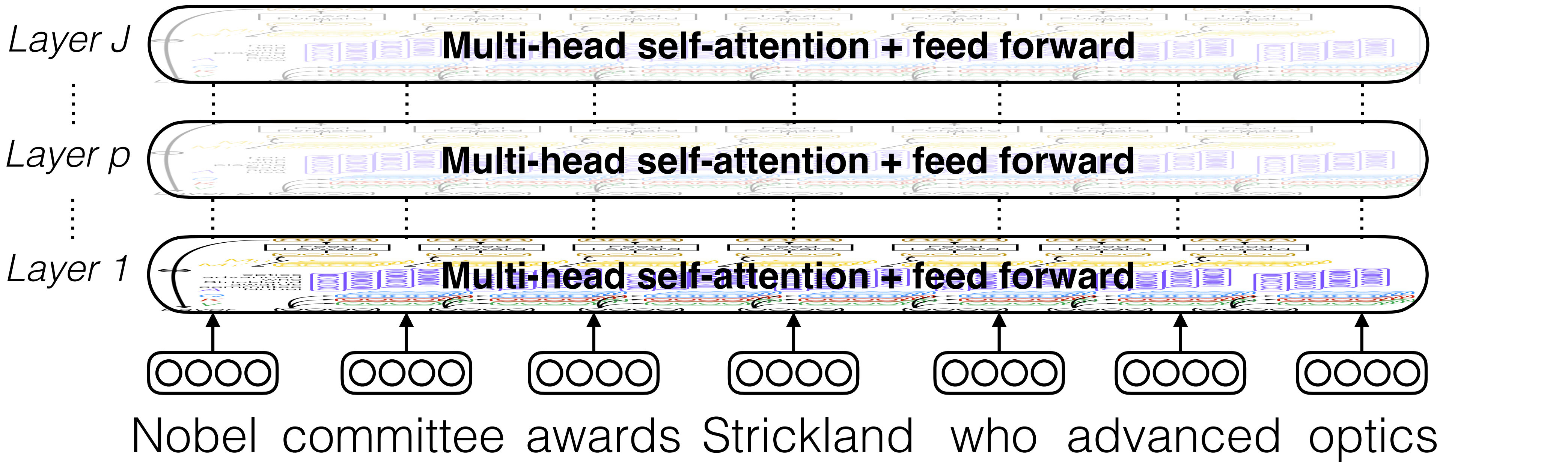
Multi-head self-attention

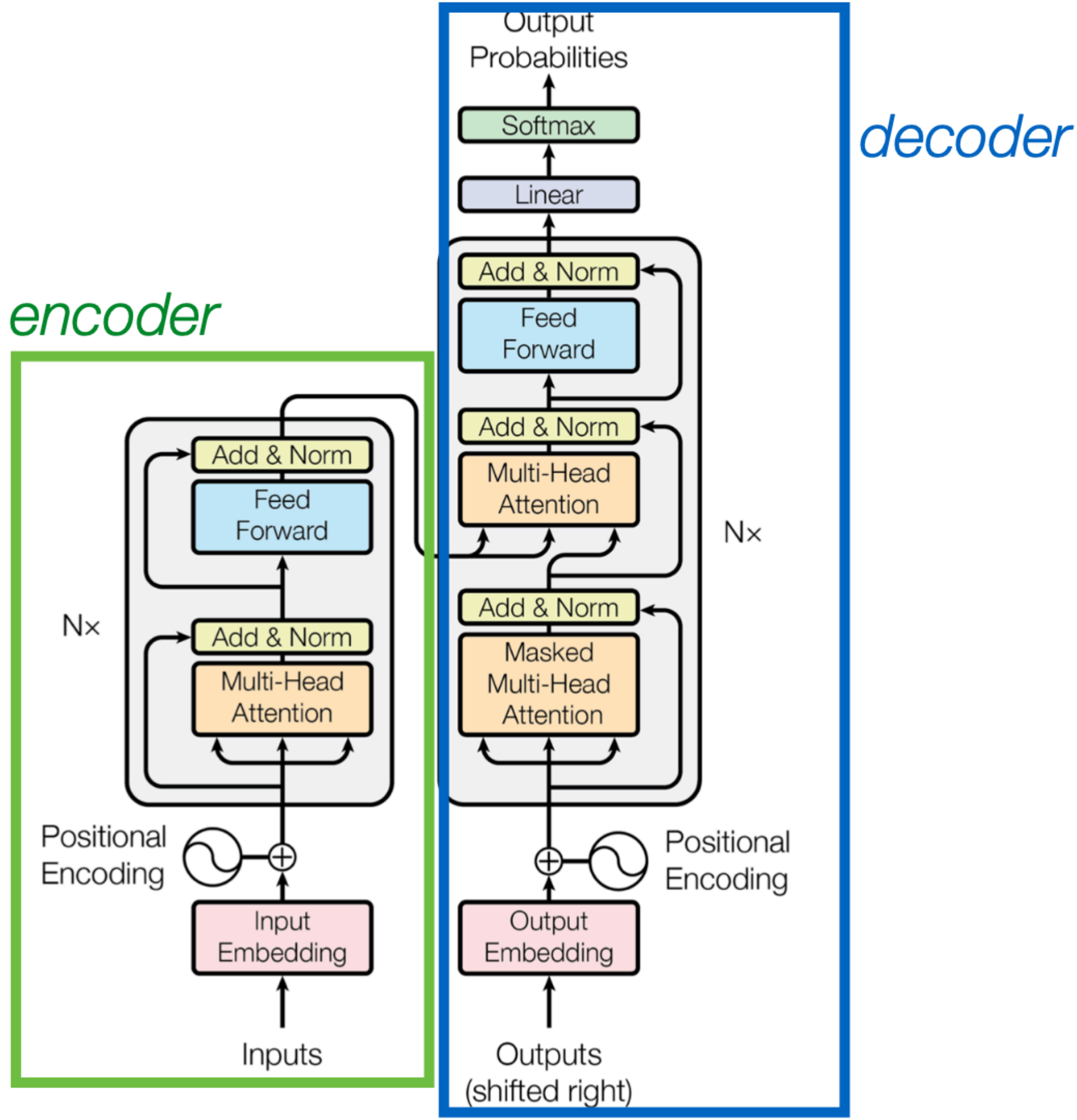


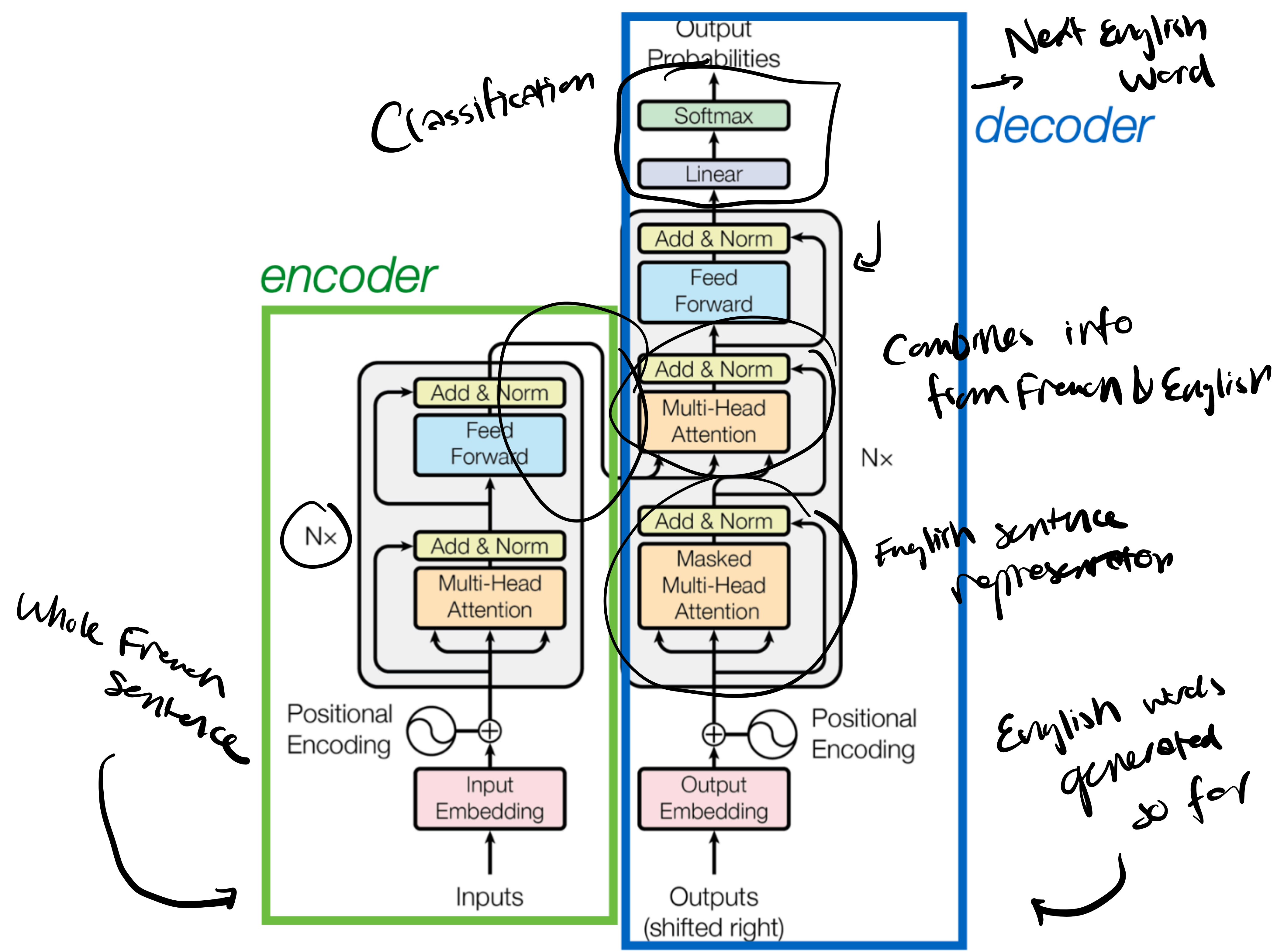
Slides by Emma Strubell!

[Vaswani et al. 2017]

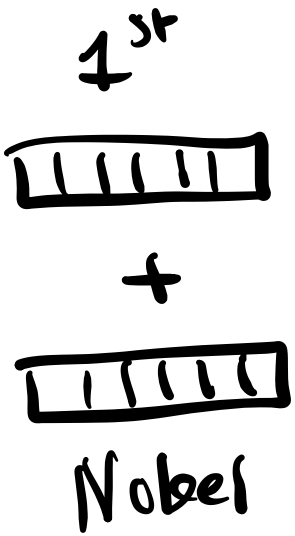
Multi-head self-attention

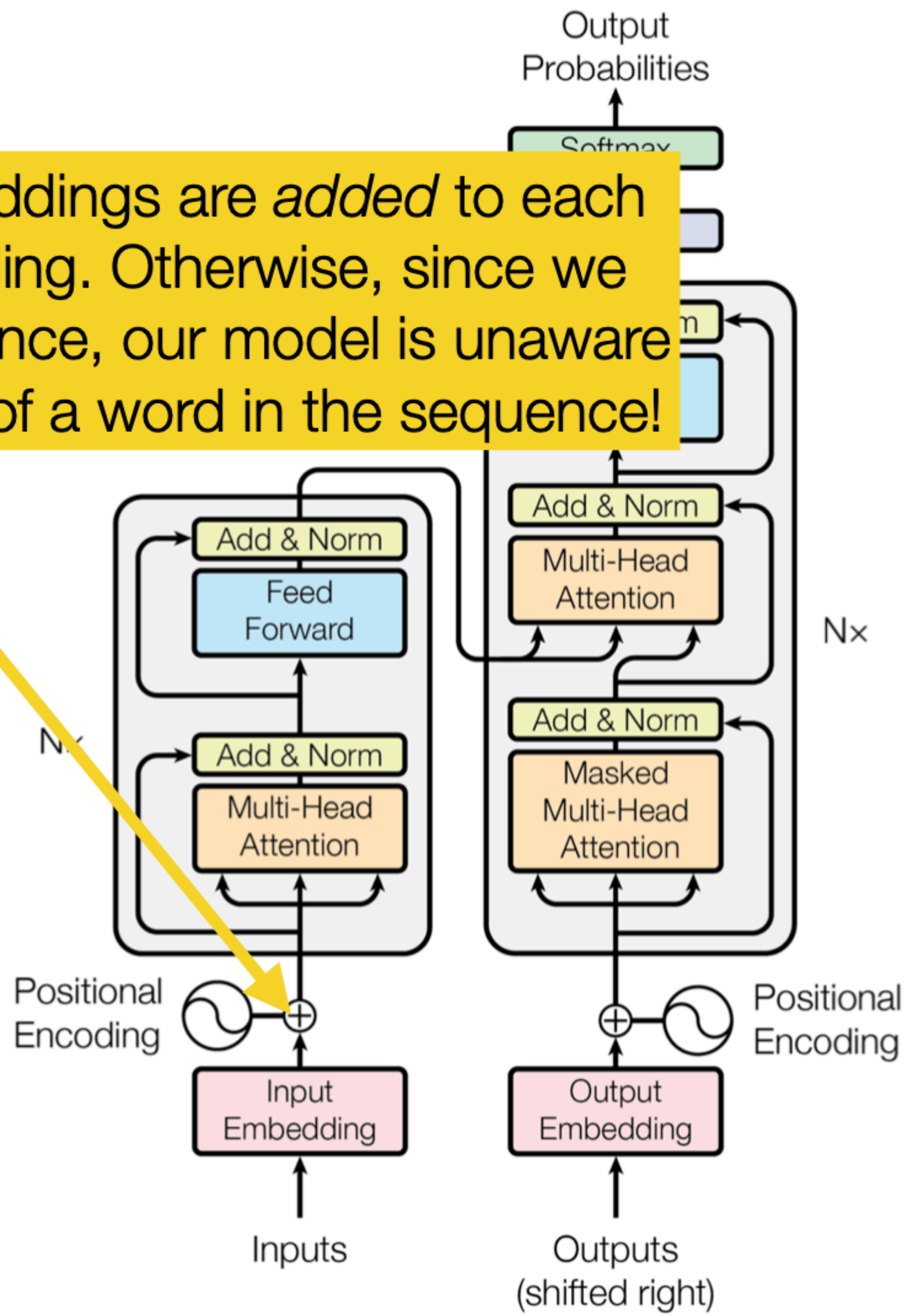



decoder
encoder
Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Input
Embedding
Inputs
Positional
Encoding
Output
Embedding
Outputs
(shifted right)

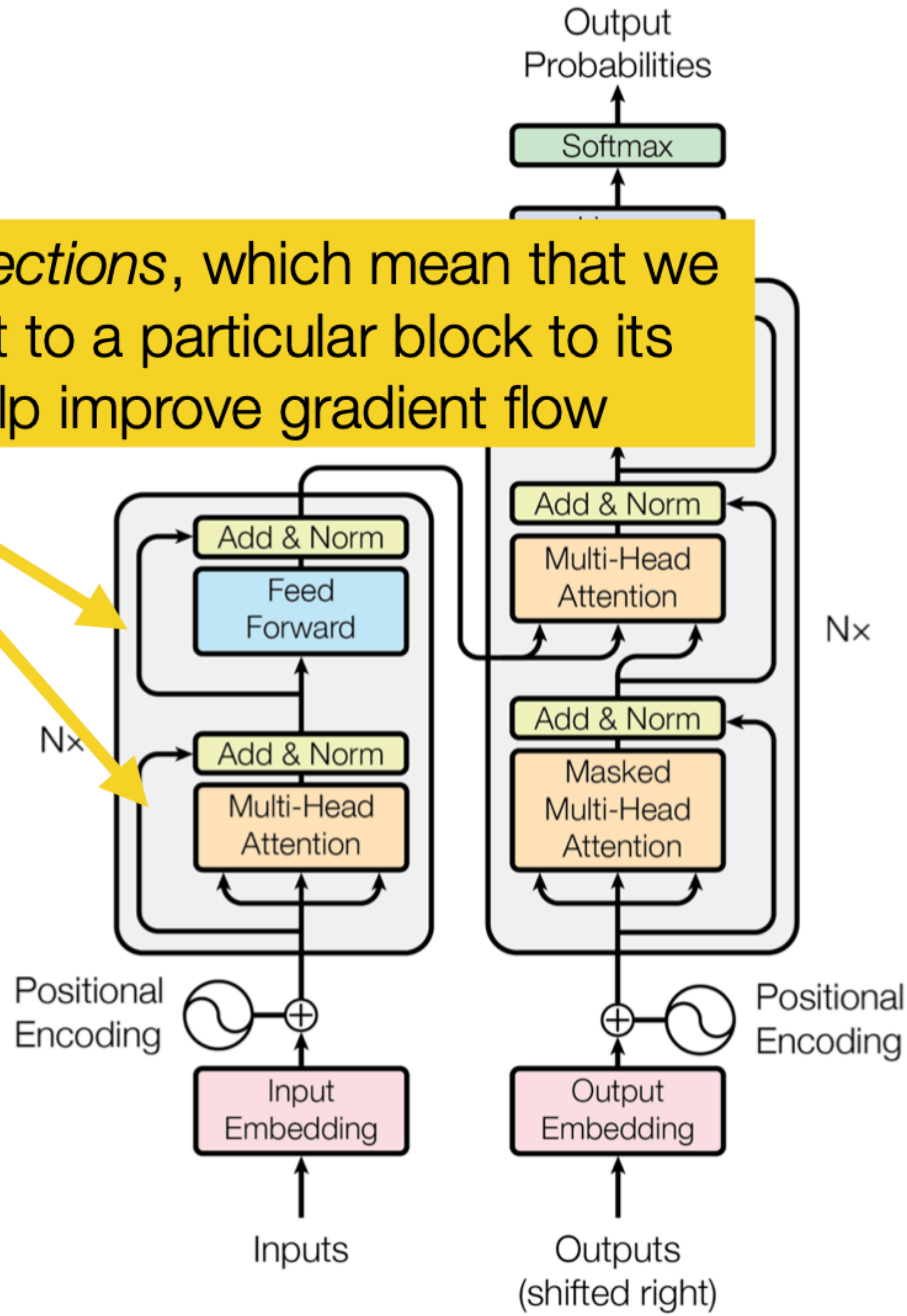

Output
Probabilities
Softmax
Linear
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Add & Norm
Masked
Multi-Head
Attention
Nx
Positional
Encoding
Output
Embedding
Outputs
(shifted right)
decoder
encoder
Add & Norm
Feed
Forward
Add & Norm
Multi-Head
Attention
Nx
Positional
Encoding
Input
Embedding
Inputs
Classification
Next English word
Combines info from French & English
English sentence representation
English words generated so far
Whole French sentence

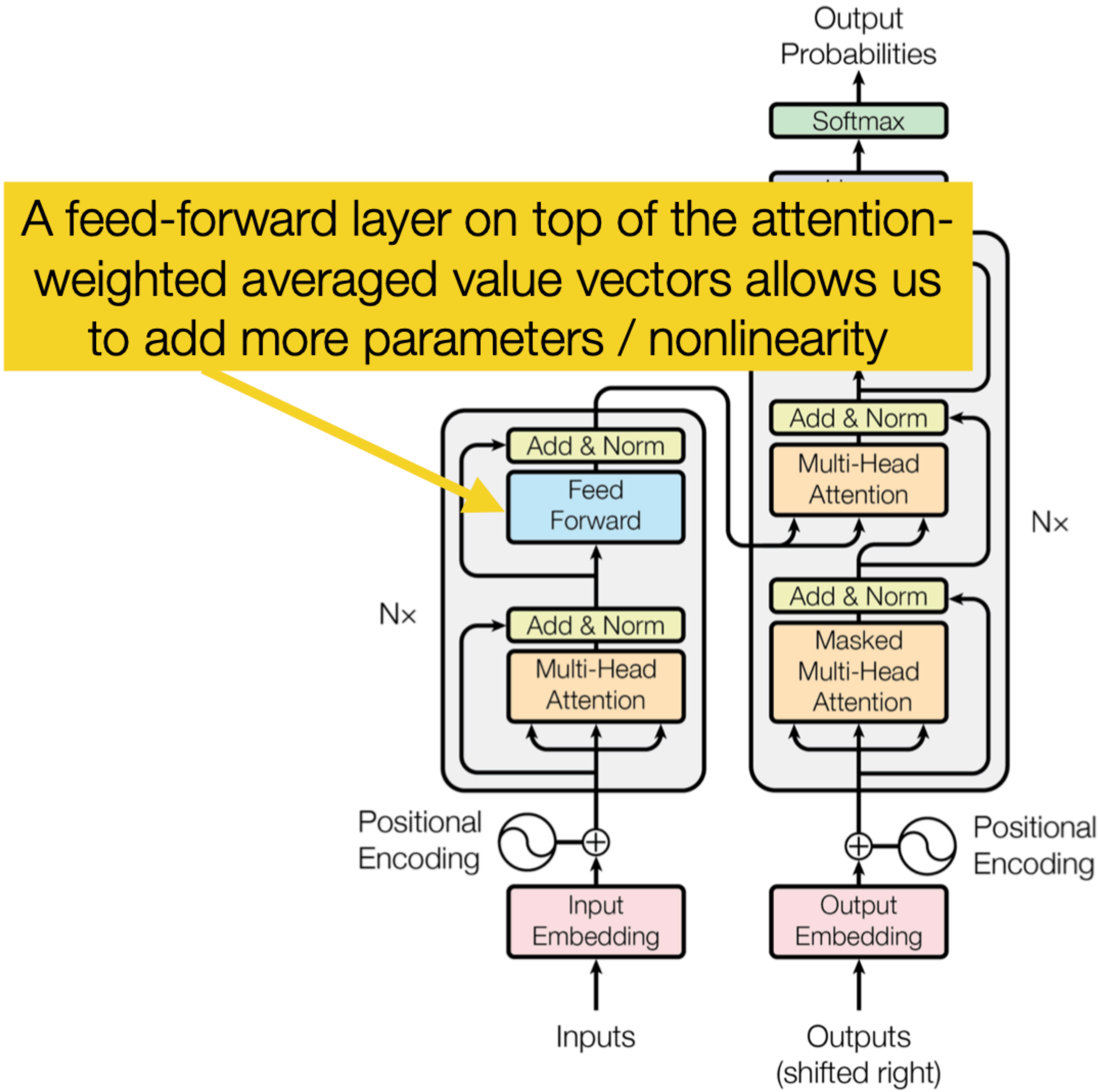
Position embeddings are *added* to each word embedding. Otherwise, since we have no recurrence, our model is unaware of the position of a word in the sequence!



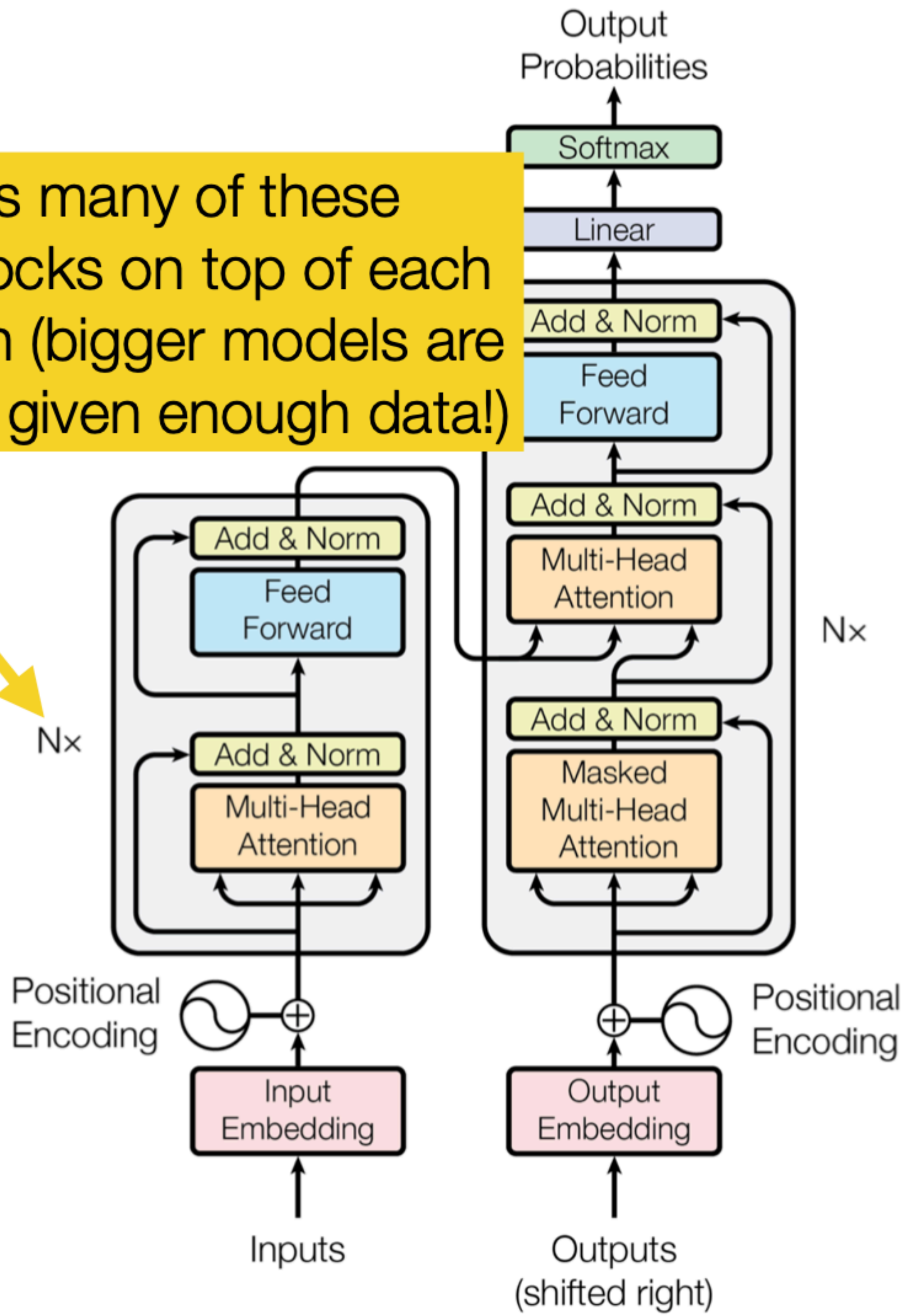


Residual connections, which mean that we add the input to a particular block to its output, help improve gradient flow

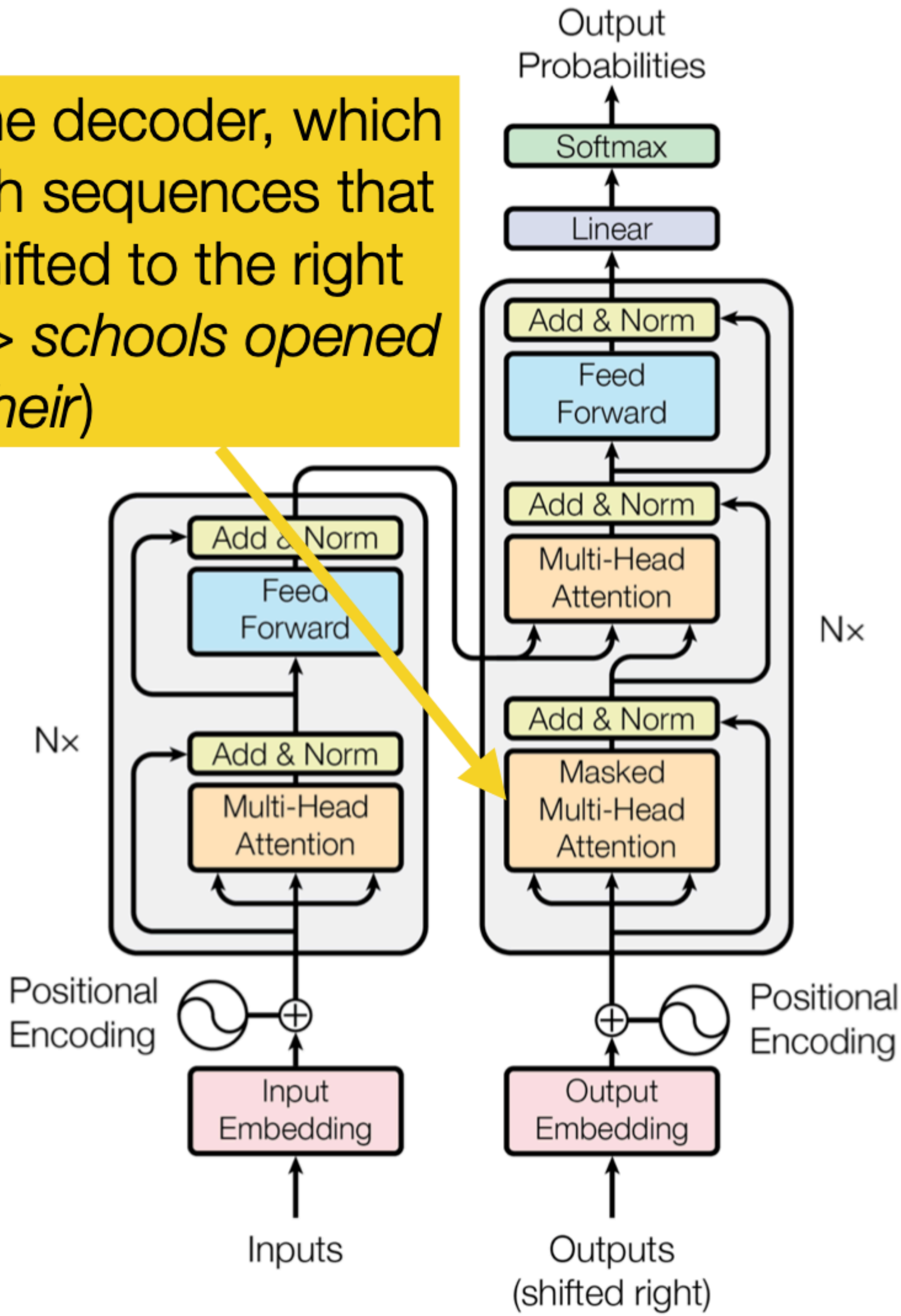



A feed-forward layer on top of the attention-weighted averaged value vectors allows us to add more parameters / nonlinearity
Output Probabilities
Softmax
Add & Norm
Multi-Head Attention
Nx
Add & Norm
Masked Multi-Head Attention
Add & Norm
Feed Forward
Nx
Add & Norm
Multi-Head Attention
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)

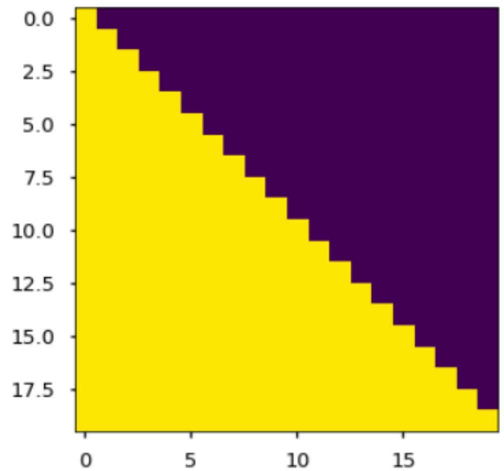
We stack as many of these *Transformer* blocks on top of each other as we can (bigger models are generally better given enough data!)

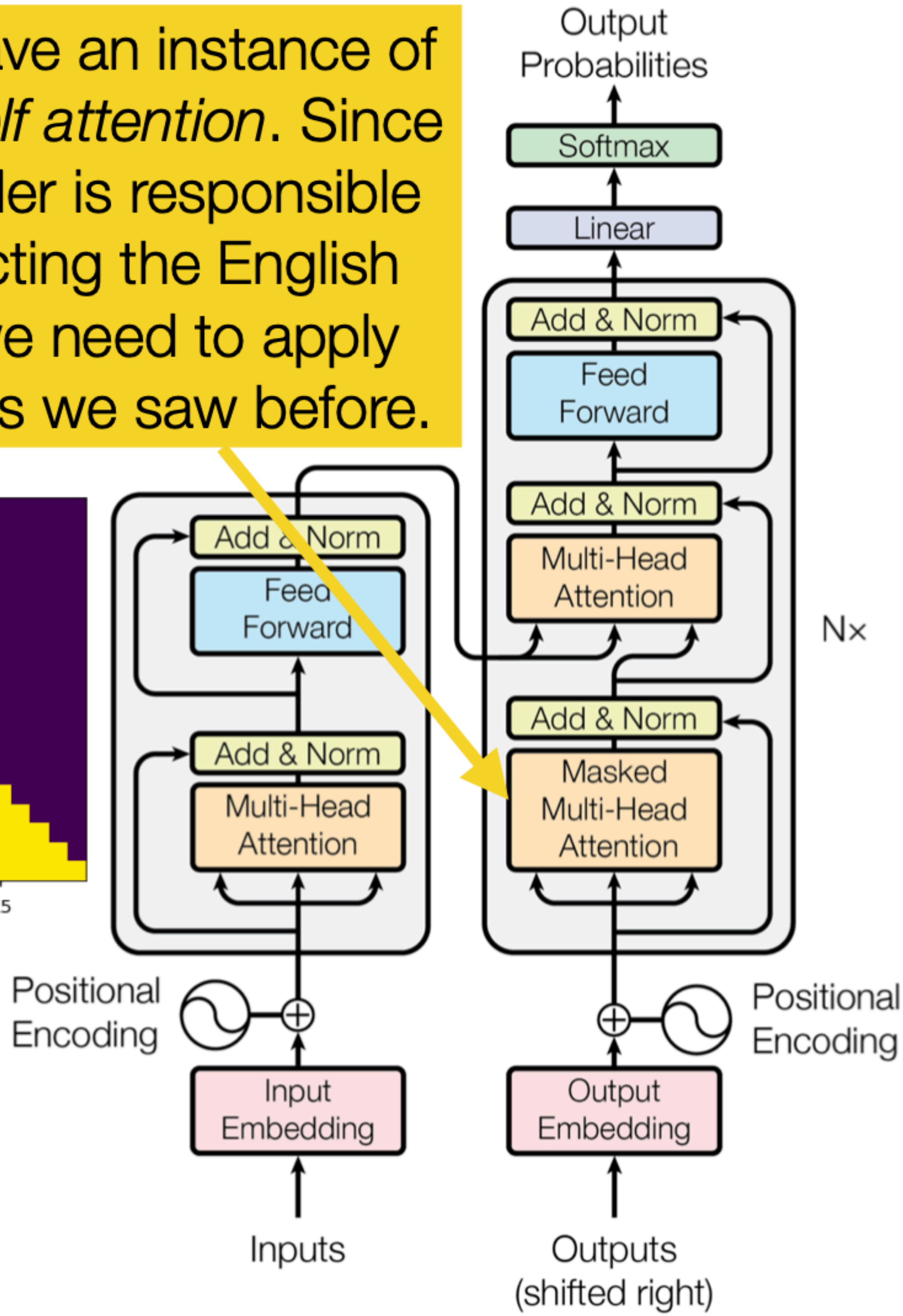


Moving onto the decoder, which takes in English sequences that have been shifted to the right (e.g., *<START> schools opened their*)



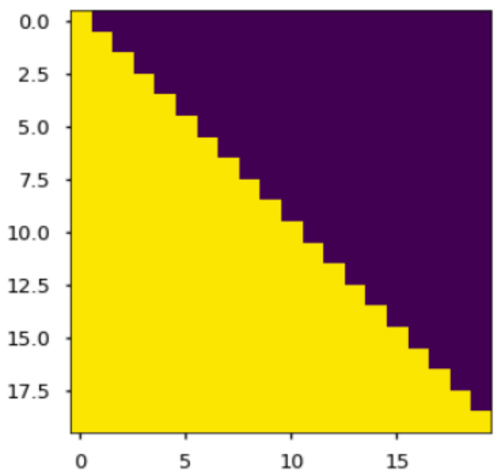
We first have an instance of *masked self attention*. Since the decoder is responsible for predicting the English words, we need to apply masking as we saw before.

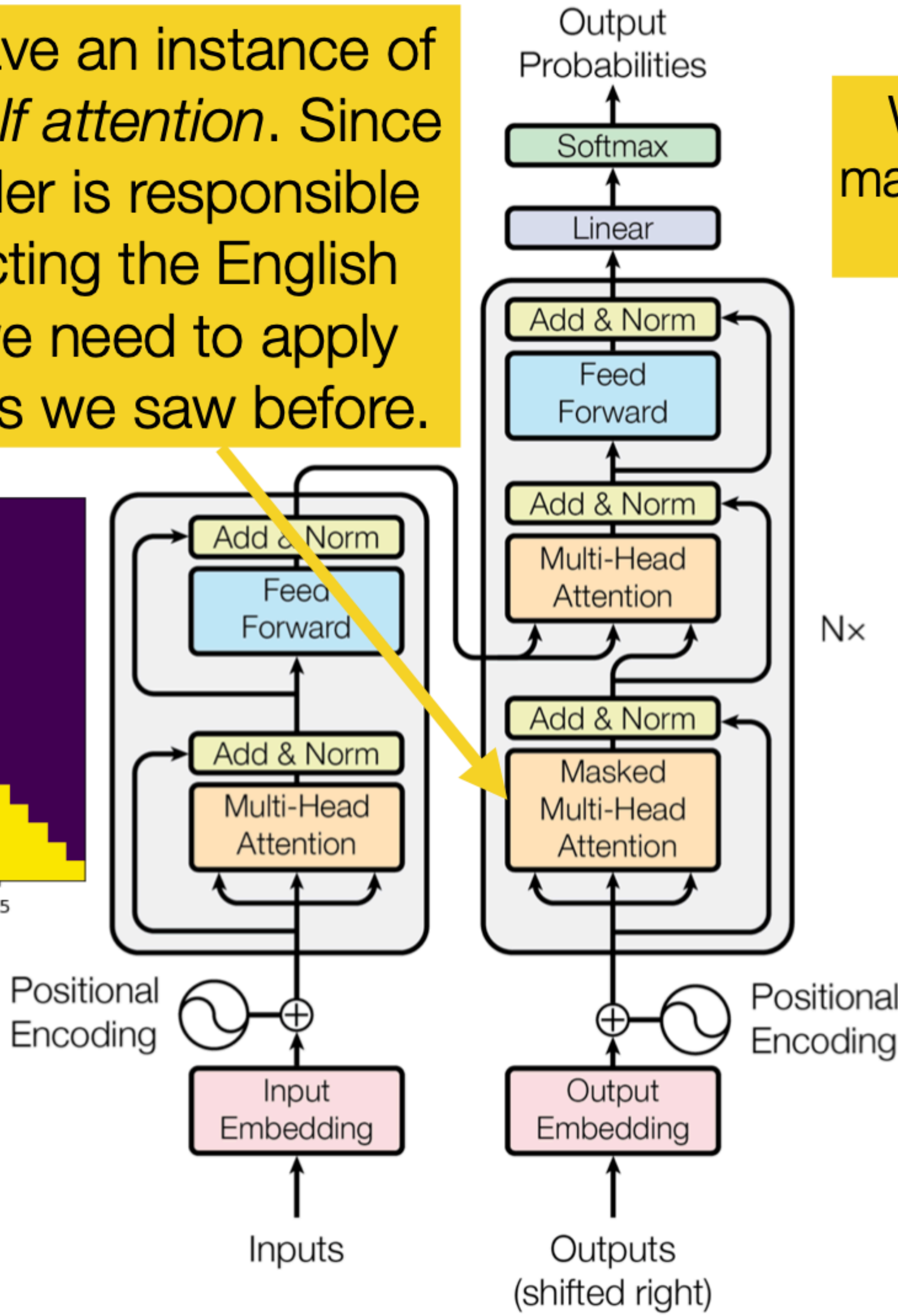




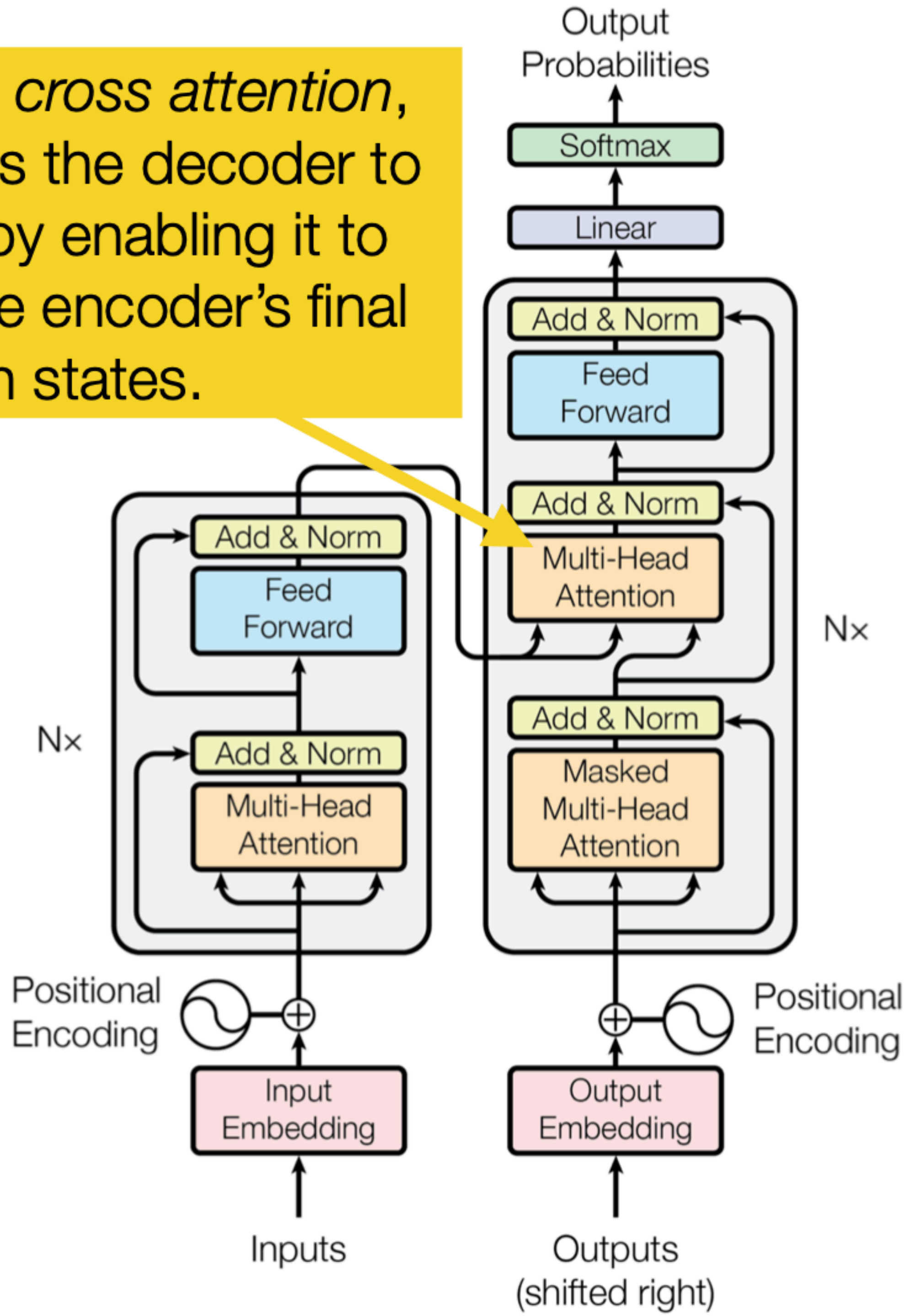
We first have an instance of *masked self attention*. Since the decoder is responsible for predicting the English words, we need to apply masking as we saw before.

Why don't we do masked self-attention in the encoder?

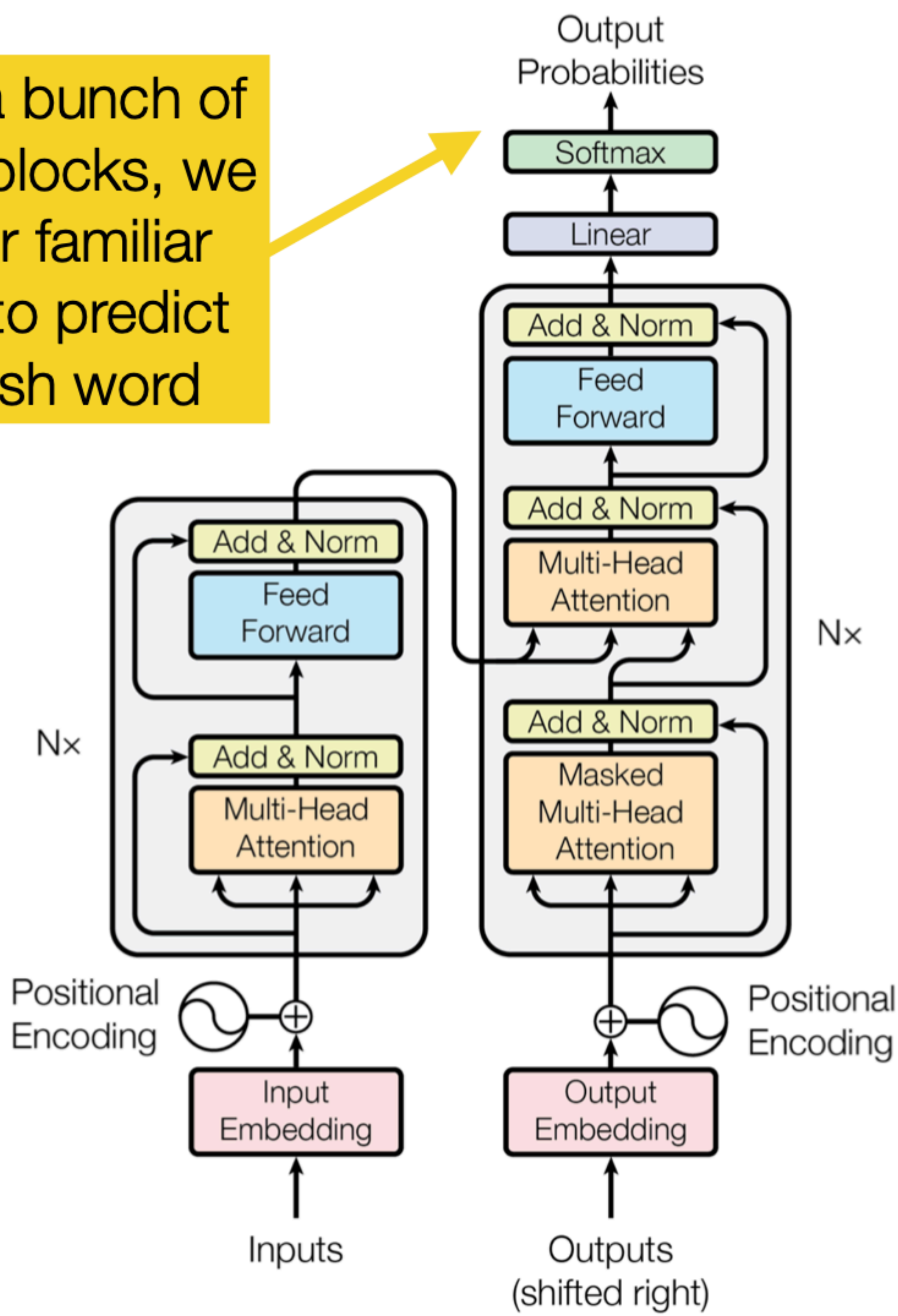




Now, we have *cross attention*, which connects the decoder to the encoder by enabling it to attend over the encoder's final hidden states.



After stacking a bunch of these decoder blocks, we finally have our familiar Softmax layer to predict the next English word



The Deep Learning Pipeline

The Deep Learning Pipeline

Deep learning models can be run in two modes:

- **Training**: update a model's weights to fit new data. This is *supervised learning* because it requires input/ output pairs (labeled data).

- **Inference**: run data through a model to make predictions. This requires only input data. It does not change the model weights.

Transfer Learning

Contemporary machine learning often involves multiple stages of training:

- **Pre-training**: train a large model that will be used by many downstream applications
  Called a foundation model in Bommasani et al. 2021

- **Fine-tuning**: adapting a pre-trained model to a new task or dataset by training it on new data, starting from existing weights.

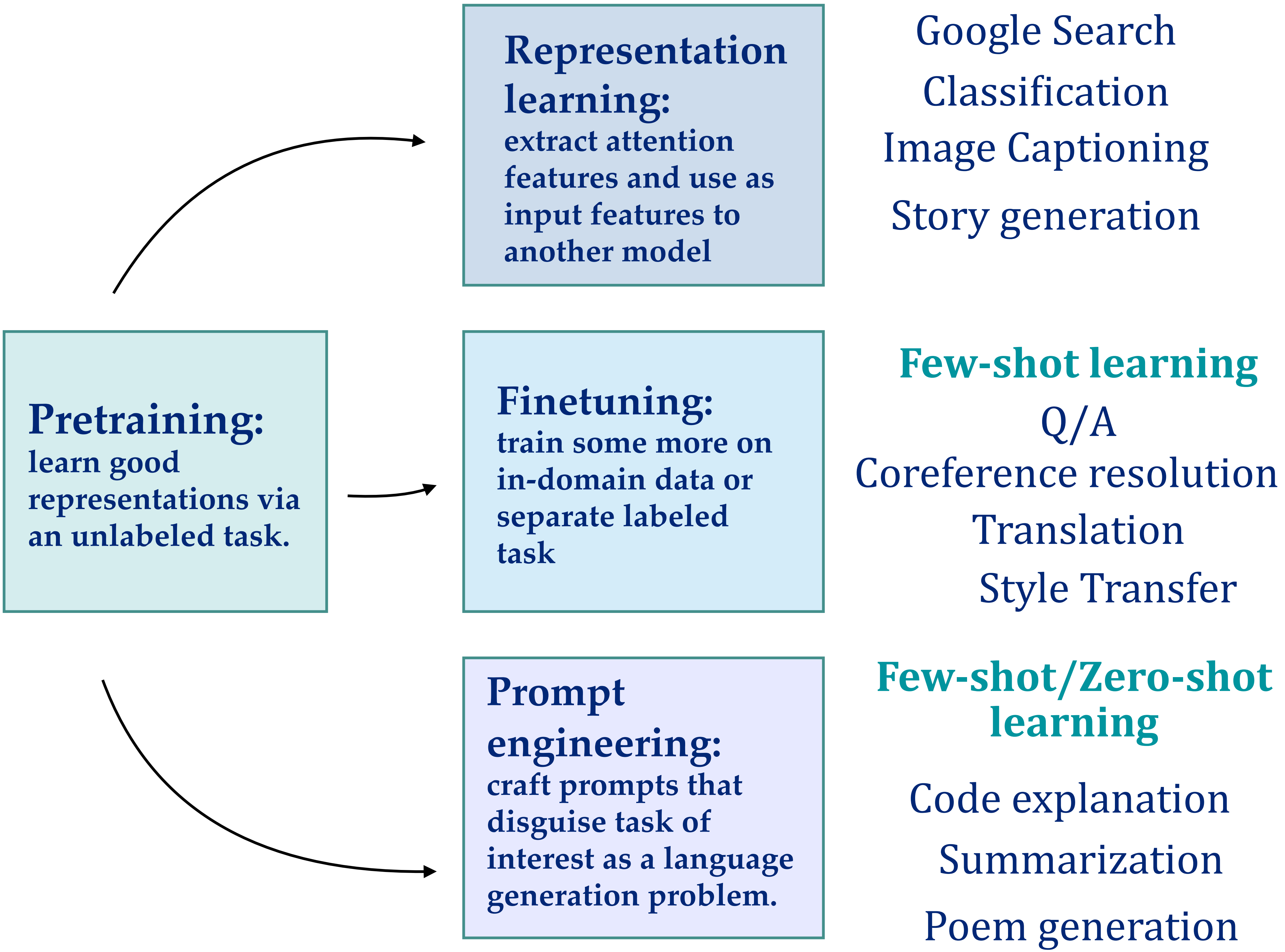
Transfer Learning

Contemporary machine learning models may also build upon other models by **freezing the weights of the original model** and taking some of its components as input.

For instance, the **weights of attention heads** may be re-used as embeddings to be fed in as input to a downstream model.

This is called **feature extraction**.

This is what we did in the recipe classifier: we took attention weights from RoBERTa to use as features in our classifier!


Representation learning:
extract attention features and use as input features to another model
Google Search
Classification
Image Captioning
Story generation
Pretraining:
learn good representations via an unlabeled task.
Finetuning:
train some more on in-domain data or separate labeled task
Few-shot learning
Q/A
Coreference resolution
Translation
Style Transfer
Prompt engineering:
craft prompts that disguise task of interest as a language generation problem.
Few-shot/Zero-shot learning
Code explanation
Summarization
Poem generation

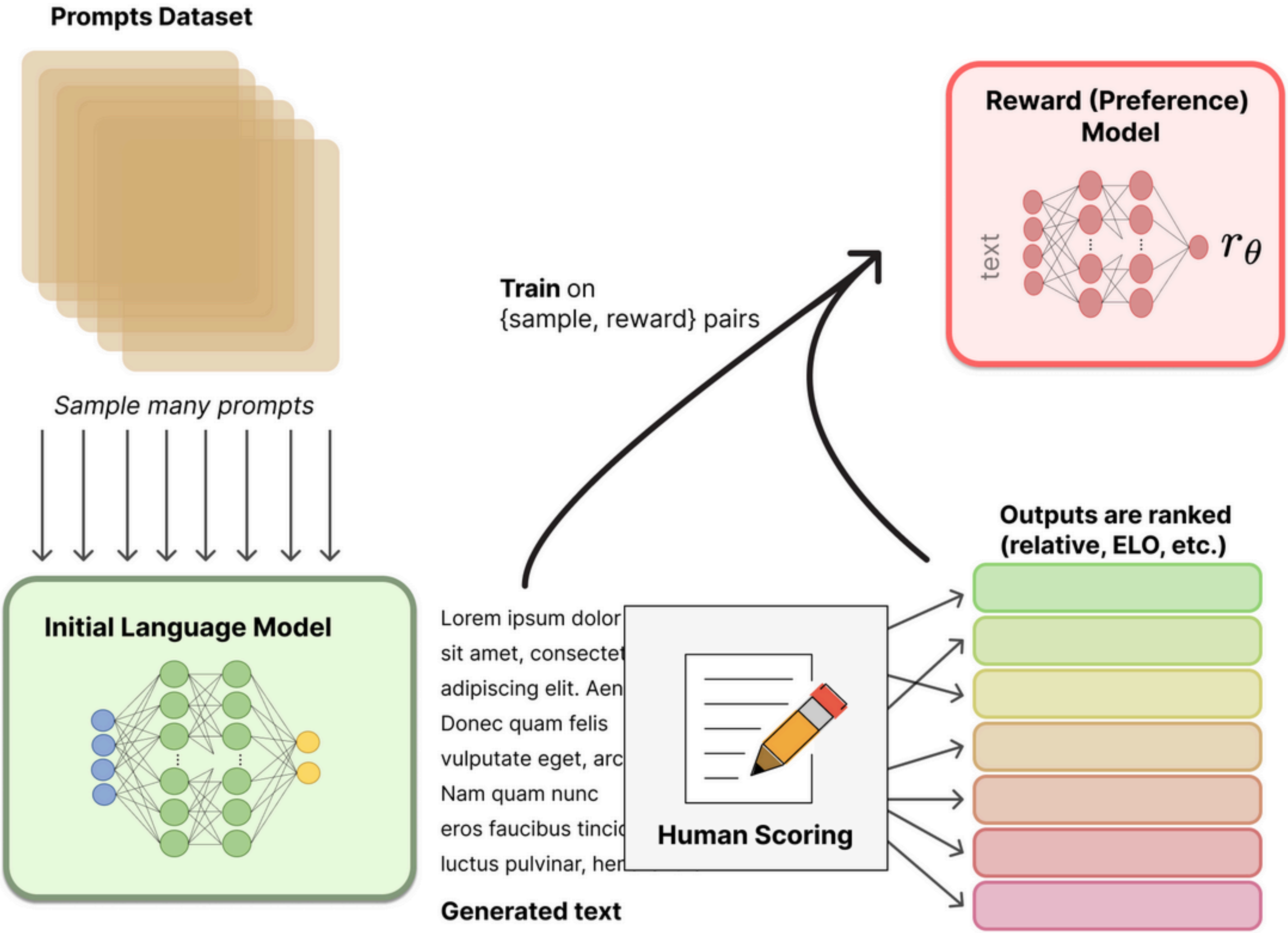
Reinforcement Learning From Human Feedback

RLHF

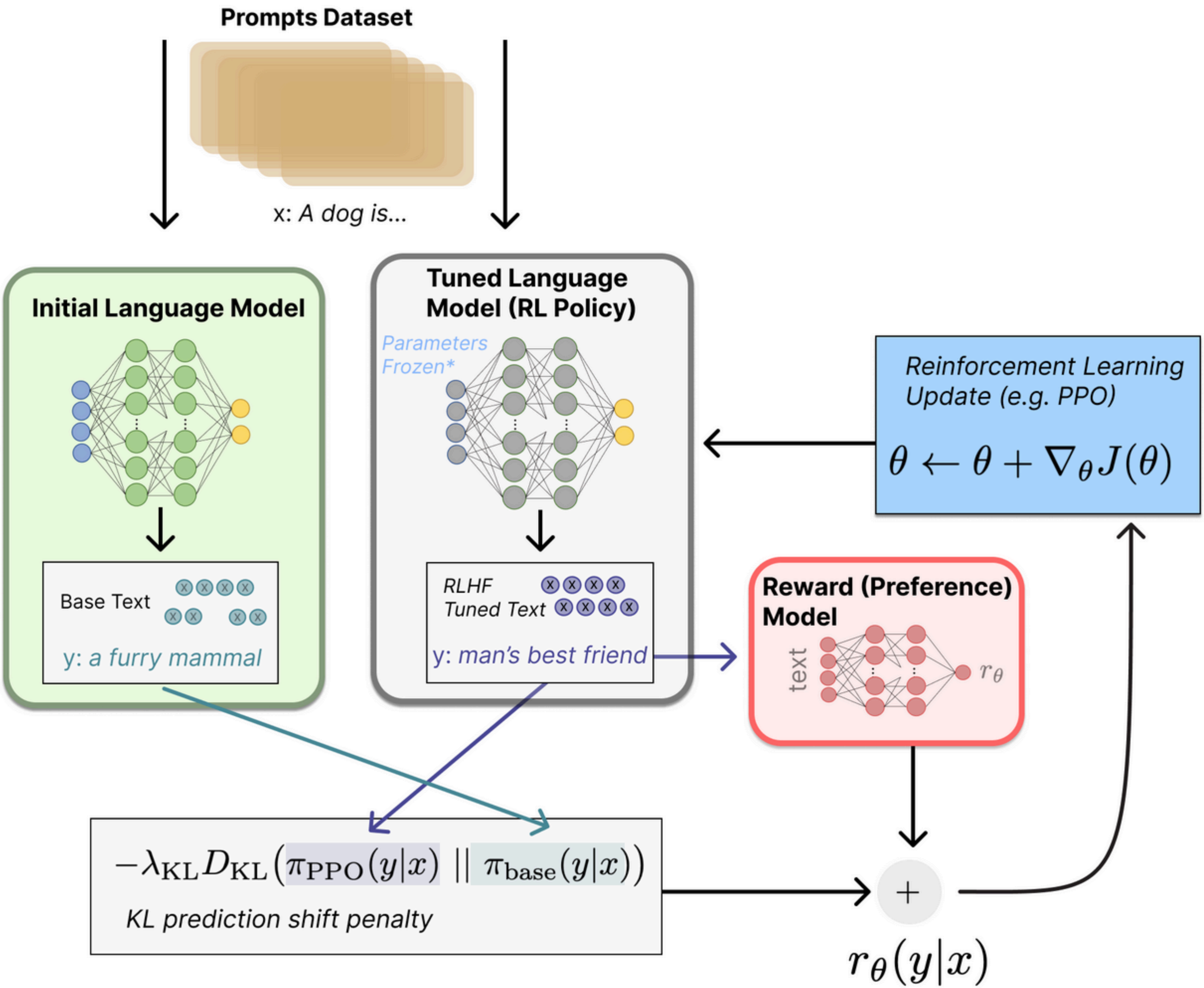
1. Pretrain your large language model
2. Train a reward model from human feedback:

text → **Reward Model** → scalar

3. Finetune (some of) your large language model using the reward model, but with a ***policy shift constraint***



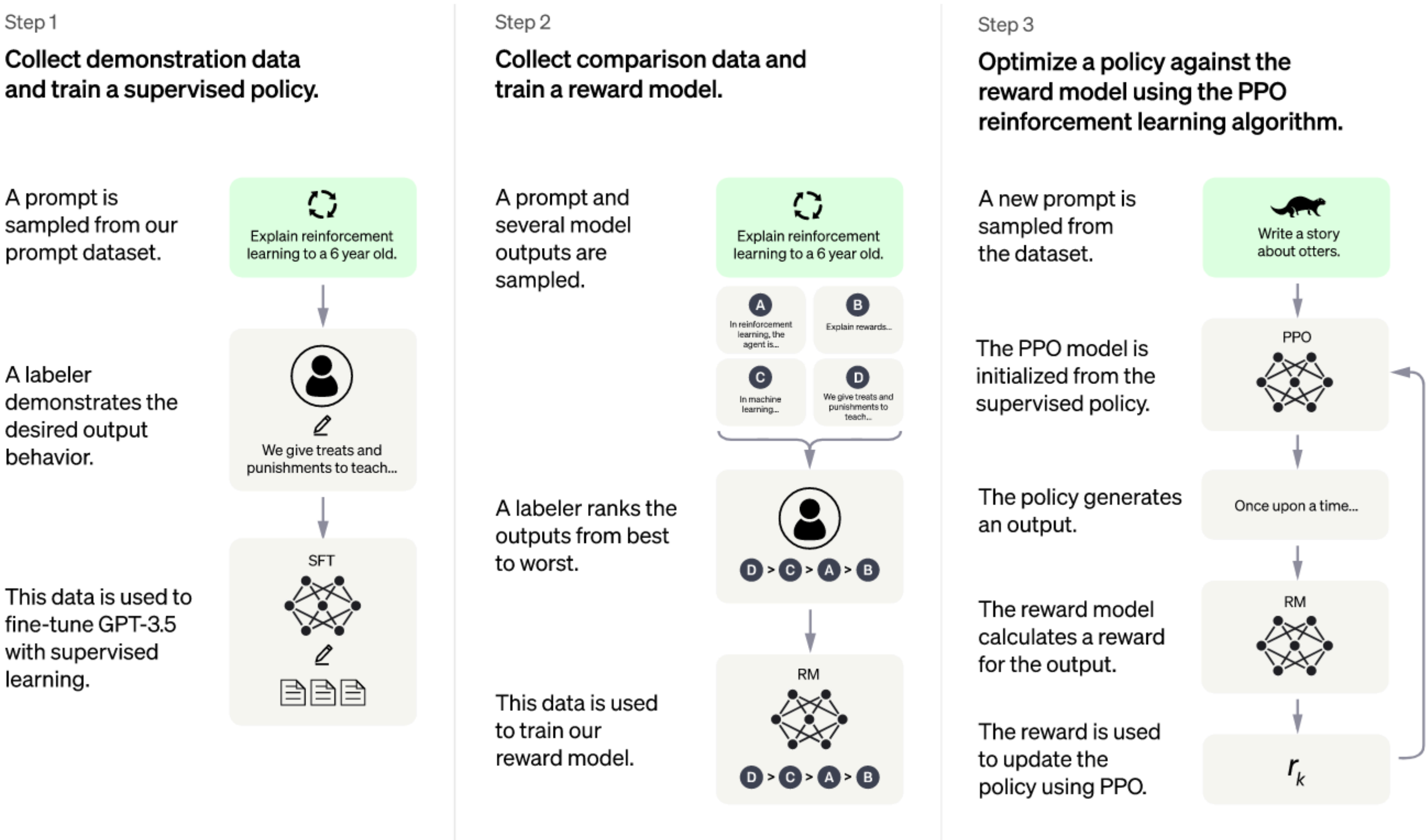
huggingface.co/blog/rlhf



huggingface.co/blog/rlhf

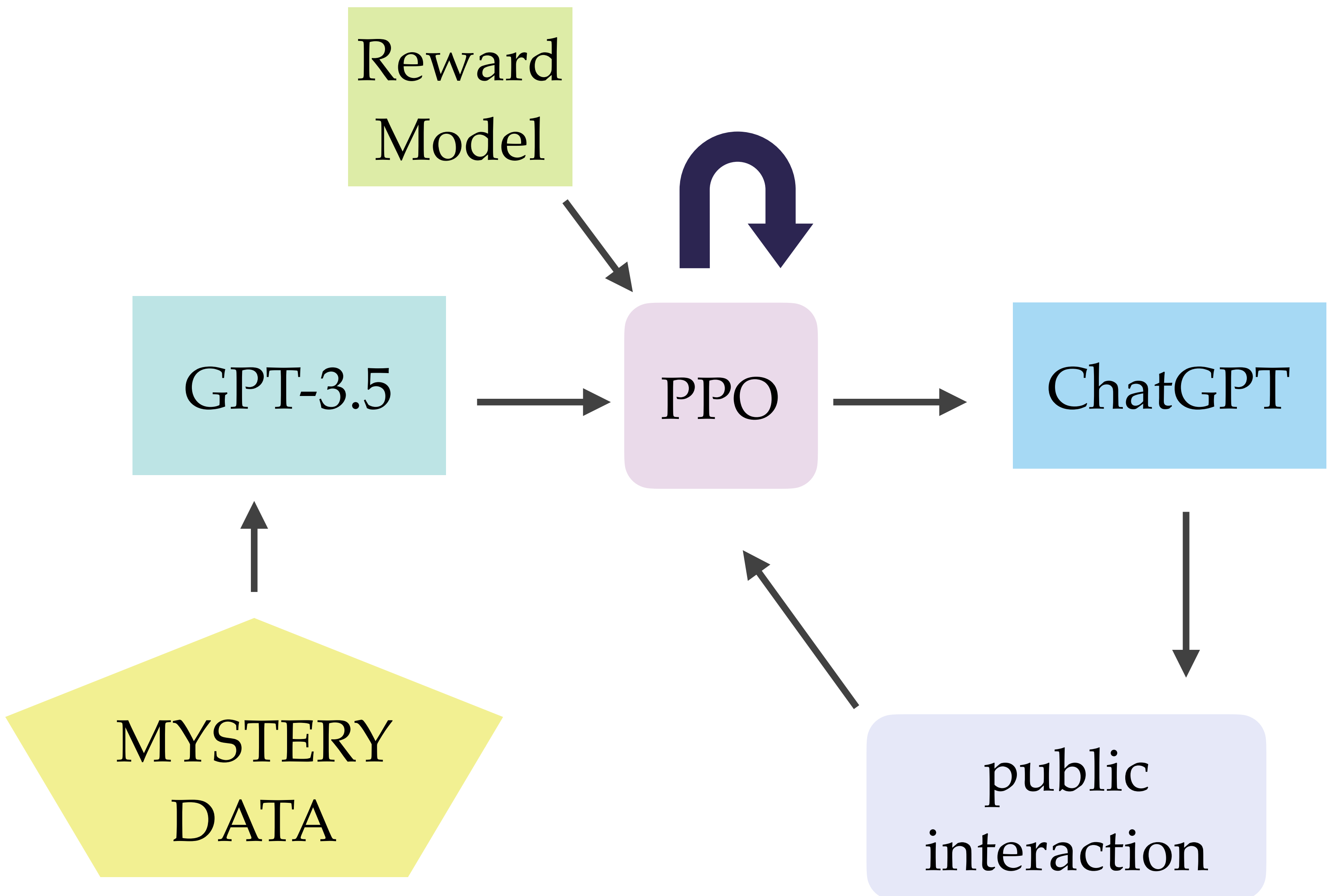
ChatGPT

Proximal Policy Optimization



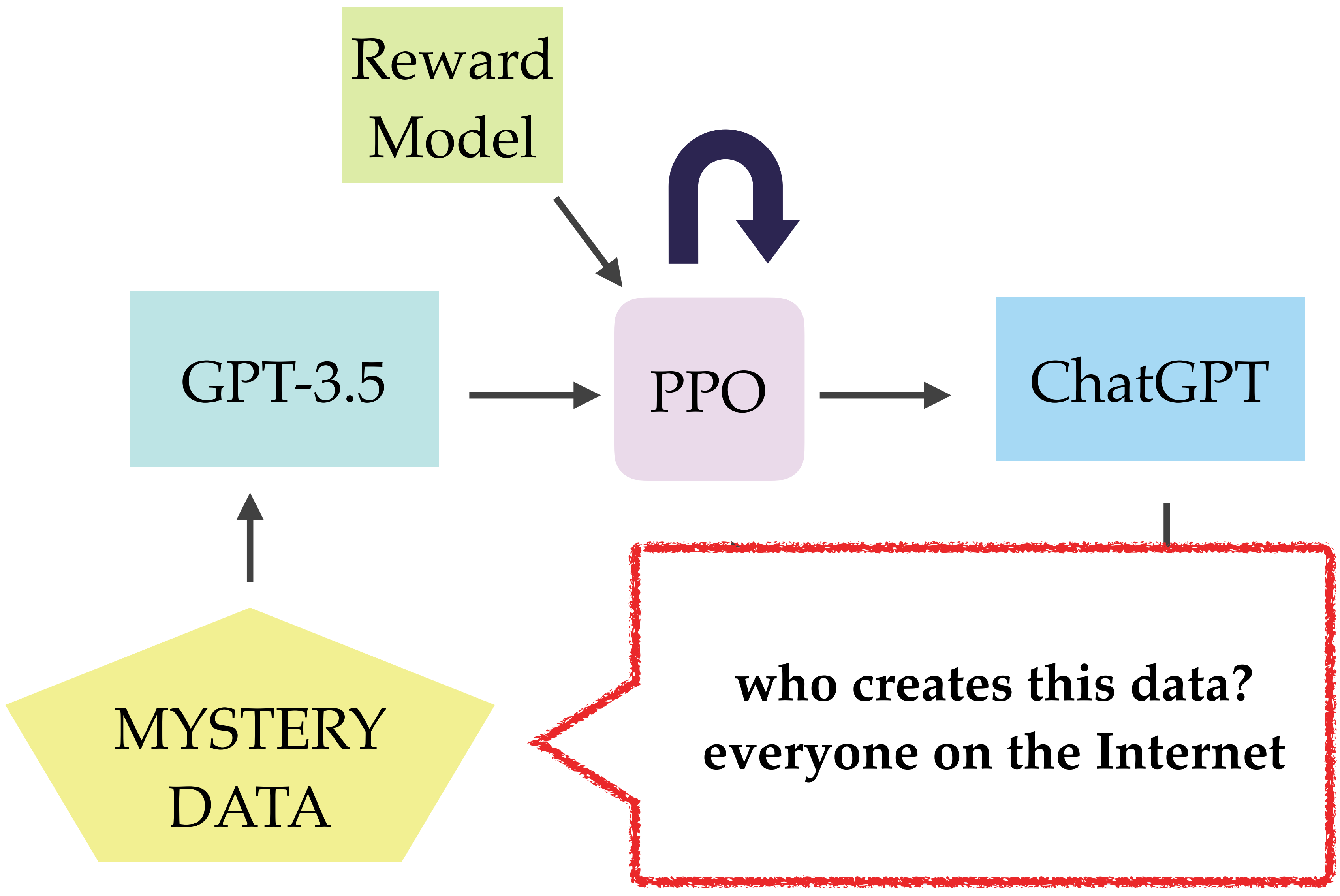
openai.com / blog / chatgpt

ChatGPT

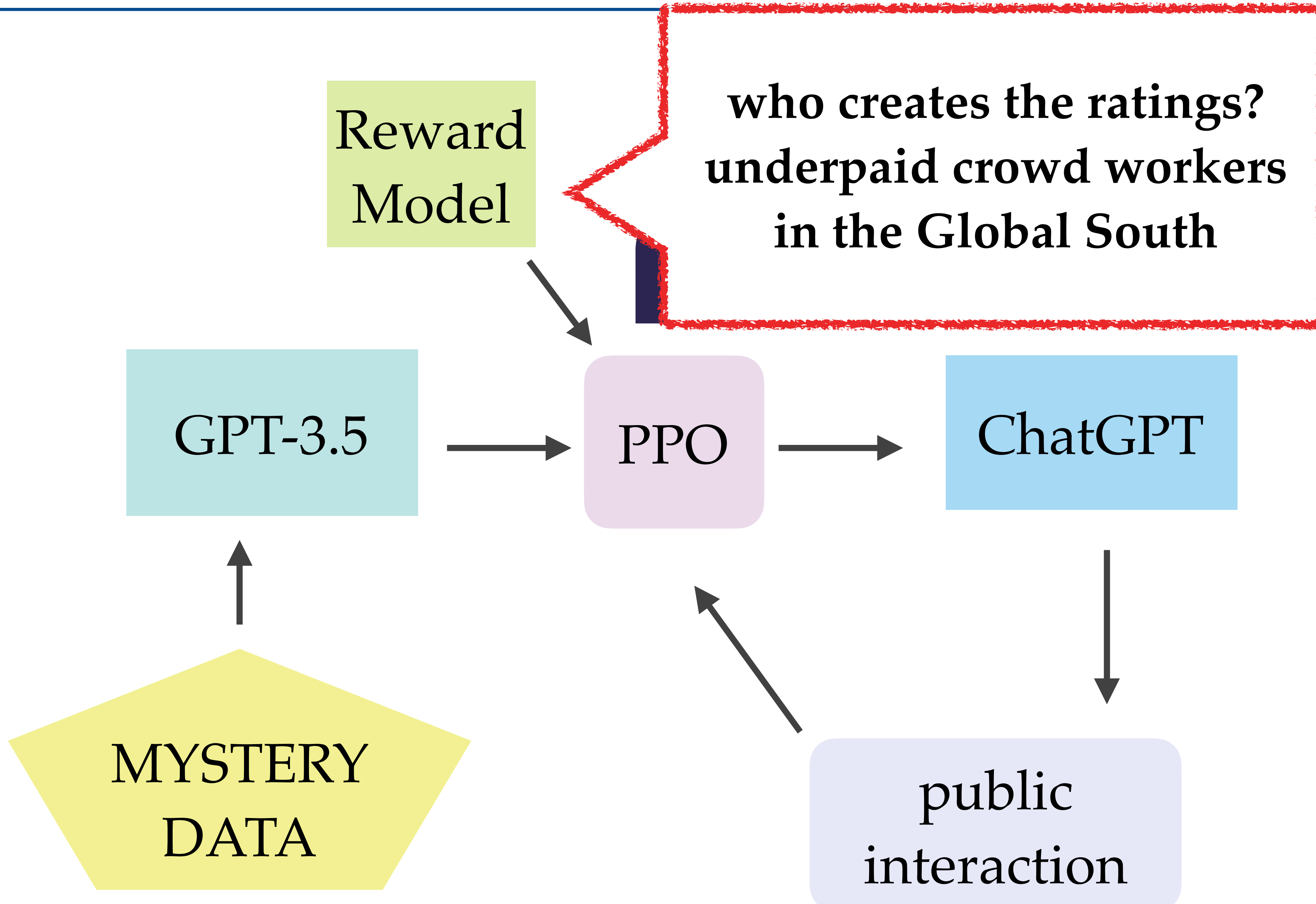


Reward Model
GPT-3.5
PPO
ChatGPT
MYSTERY DATA
public interaction

ChatGPT


Reward Model
GPT-3.5
PPO
ChatGPT
MYSTERY DATA
who creates this data?
everyone on the Internet

ChatGPT



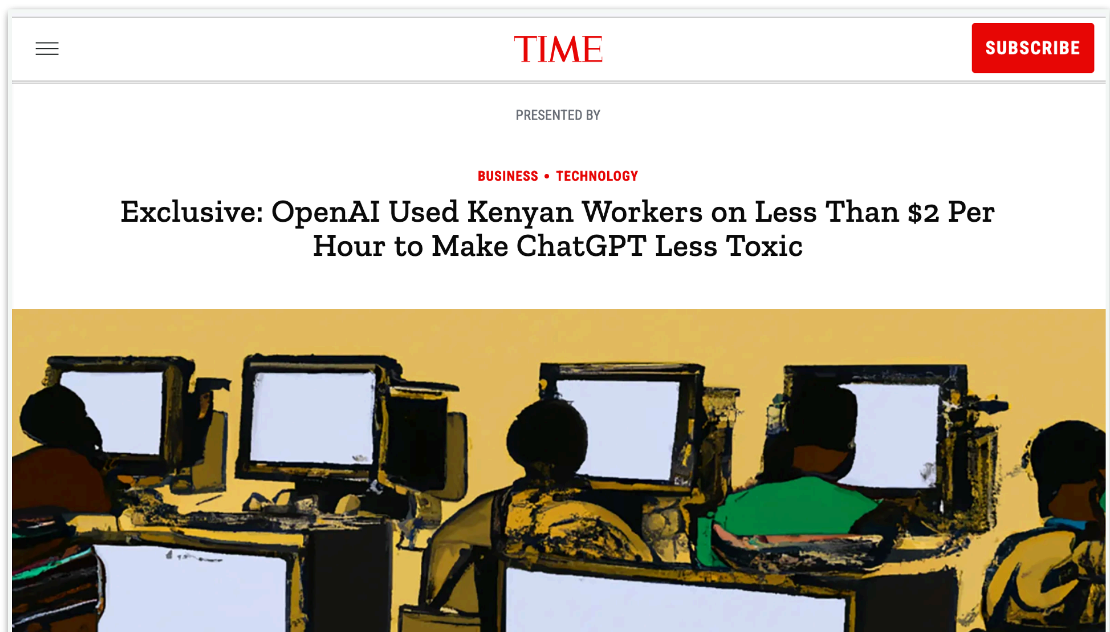
Reward Model
who creates the ratings?
underpaid crowd workers
in the Global South
GPT-3.5
PPO
ChatGPT
MYSTERY DATA
public interaction

ChatGPT

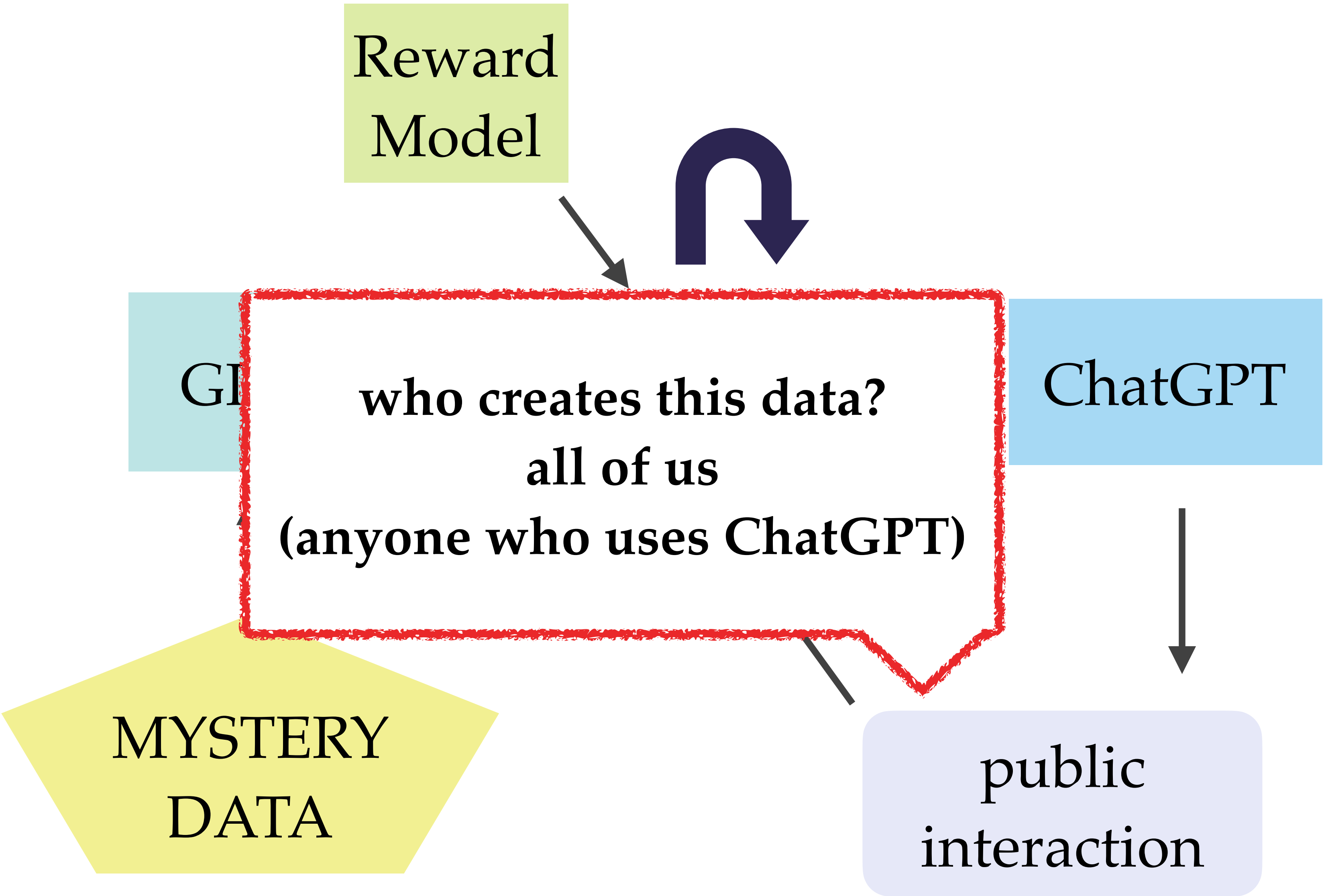
"OpenAI sent tens of thousands of snippets of text to an outsourcing firm in Kenya, beginning in November 2021. Much of that text appeared to have been pulled from the darkest recesses of the internet. Some of it described situations in graphic detail like child sexual abuse, bestiality, murder, suicide, torture, self harm, and incest."

"OpenAI's outsourcing partner in Kenya was Sama, a San Francisco-based firm. Sama markets itself as an "ethical AI" company."

"The data labelers employed by Sama on behalf of OpenAI were paid a take-home wage of between around $1.32 and $2 per hour depending on seniority and performance."



ChatGPT


Reward Model
GI
ChatGPT
who creates this data?
all of us
(anyone who uses ChatGPT)
MYSTERY DATA
public interaction

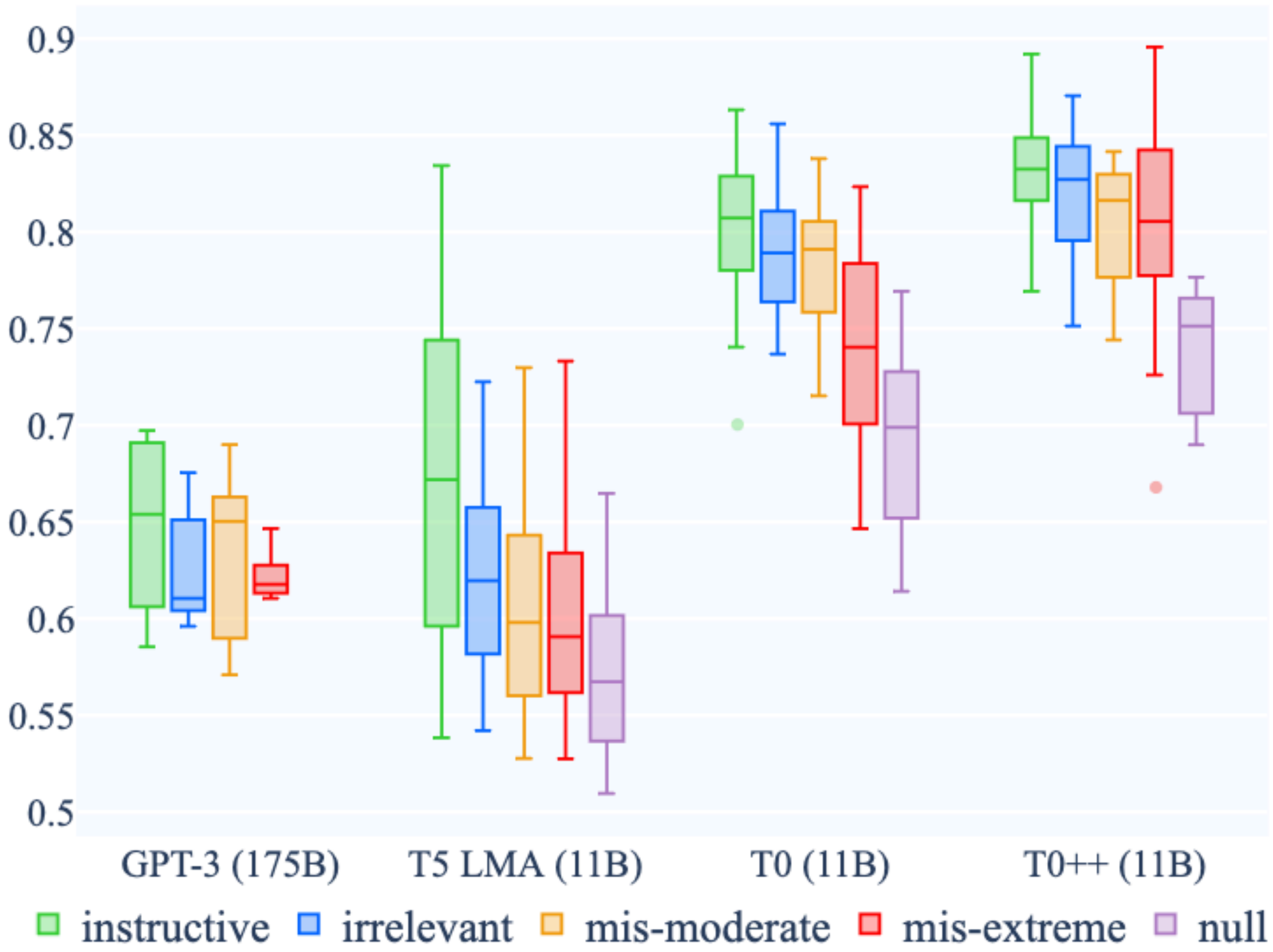
Prompt Engineering

Chain-of-Thought Reasoning

One idea is to make the model generate reasoning before an answer. This guarantees that the answer is conditioned on the reasoning. Some people think this could improve the quality of the answer. However, other work has shown that the answer is not always consistent with the given reasoning.

```
Question: Tom and Elizabeth have a competition to climb a
hill. Elizabeth takes 30 minutes to climb the hill. Tom
takes four times as long as Elizabeth does to climb the
hill. How many hours does it take Tom to climb up the hill?
Answer: It takes Tom 30*4 = <<30*4=120>>120 minutes to
climb the hill.
It takes Tom 120/60 = <<120/60=2>>2 hours to climb the
hill.
So the answer is 2.
===
Question: Jack is a soccer player. He needs to buy two
pairs of socks and a pair of soccer shoes. Each pair of
socks cost $9.50, and the shoes cost $92. Jack has $40.
How much more money does Jack need?
Answer: The total cost of two pairs of socks is $9.50 x 2
= $<<9.5*2=19>>19.
The total cost of the socks and the shoes is $19 + $92 =
$<<19+92=111>>111.
Jack need $111 - $40 = $<<111-40=71>>71 more.
So the answer is 71.
===
Question: Marty has 100 centimeters of ribbon that he must
cut into 4 equal parts. Each of the cut parts must be
divided into 5 equal parts. How long will each final cut
be?
```

What Are Prompts Really Doing?



Results from Webson & Pavlick (2022)

Does CoT Help?

Solving and Generating NPR Sunday Puzzles with Large Language Models

Jingmiao Zhao and Carolyn Jane Anderson
Computer Science Department
Wellesley College
Wellesley, MA 02482 USA
carolyn.anderson@wellesley.edu

Abstract

We explore the ability of large language models to solve and generate puzzles from the NPR Sunday Puzzle game show using PUZZLEQA, a dataset comprising 15 years of on-air puzzles. We evaluate four large language models using PUZZLEQA, in both multiple choice and free response formats, and explore two prompt engineering techniques to improve free response performance: chain-of-thought reasoning and prompt summarization. We find that state-of-the-art large language models can solve many PUZZLEQA puzzles: the best model, GPT-3.5, achieves 50.2% loose accuracy. However, in our few-shot puzzle generation experiment, we find no evidence that models can generate puzzles: GPT-3.5 generates puzzles with answers that do not conform to the generated rules. Puzzle generation remains a challenging task for future work.

Puzzle Description: Today's puzzle involves "consonyms," which are words that have the same consonants in the same order but with different vowels. Every answer is the name of a country.
Question: MINGLE
Answer: MONGOLIA

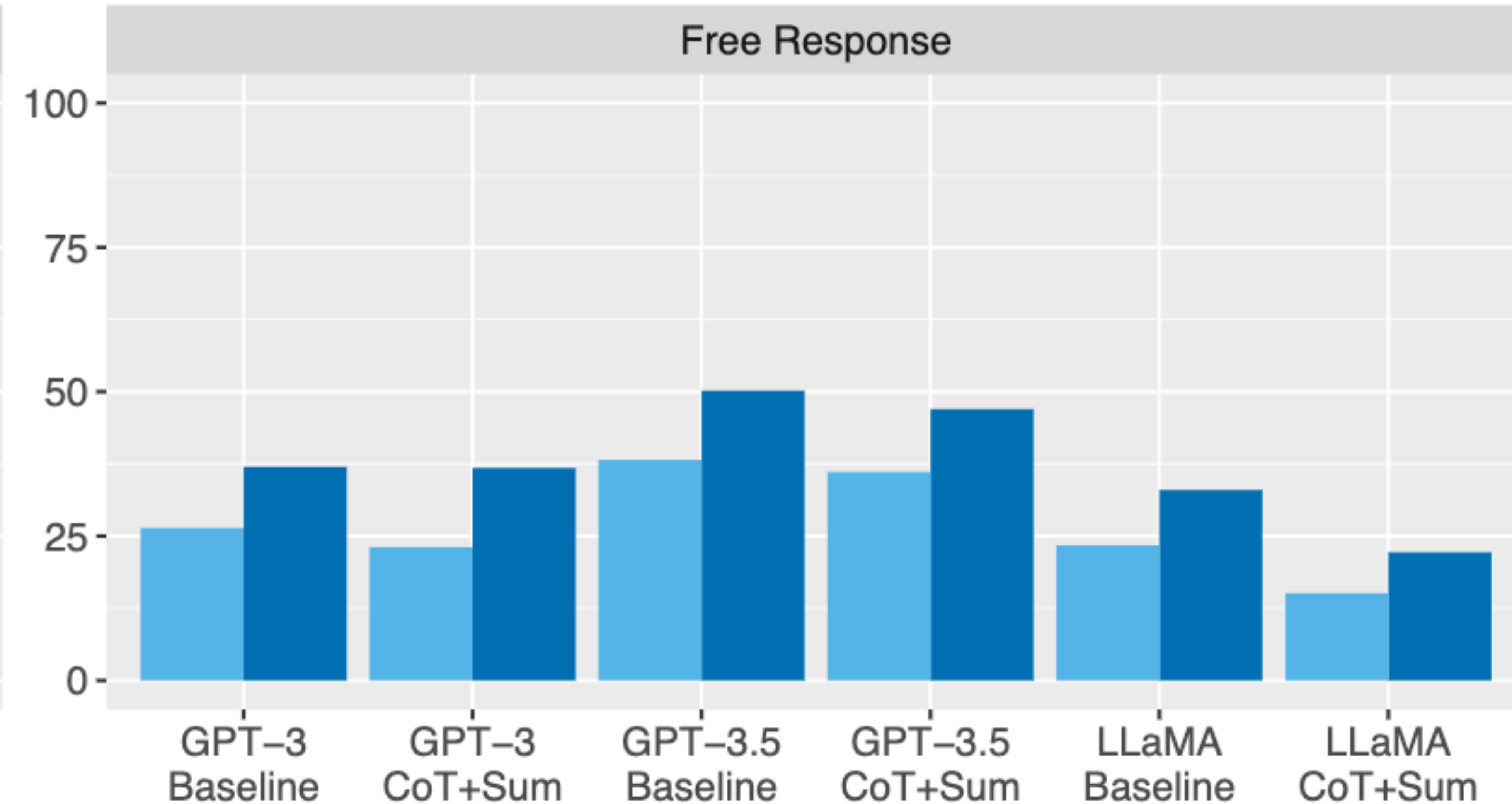
Figure 1: NPR Sunday Puzzle from March 12, 2023

Benchmarking AI through Games

Our work continues the tradition of evaluating AI progress through puzzles and games (Ferrucci 2012; Rodriguez et al. 2021; Rozner, Potts, and Mahowald 2021; Sobieszek and Price 2022). Contemporary LLMs have demonstrated strong

Does CoT Help?

Maybe not?



Continuous Prompting

Humans write discrete prompts, which are then turned into text embeddings.

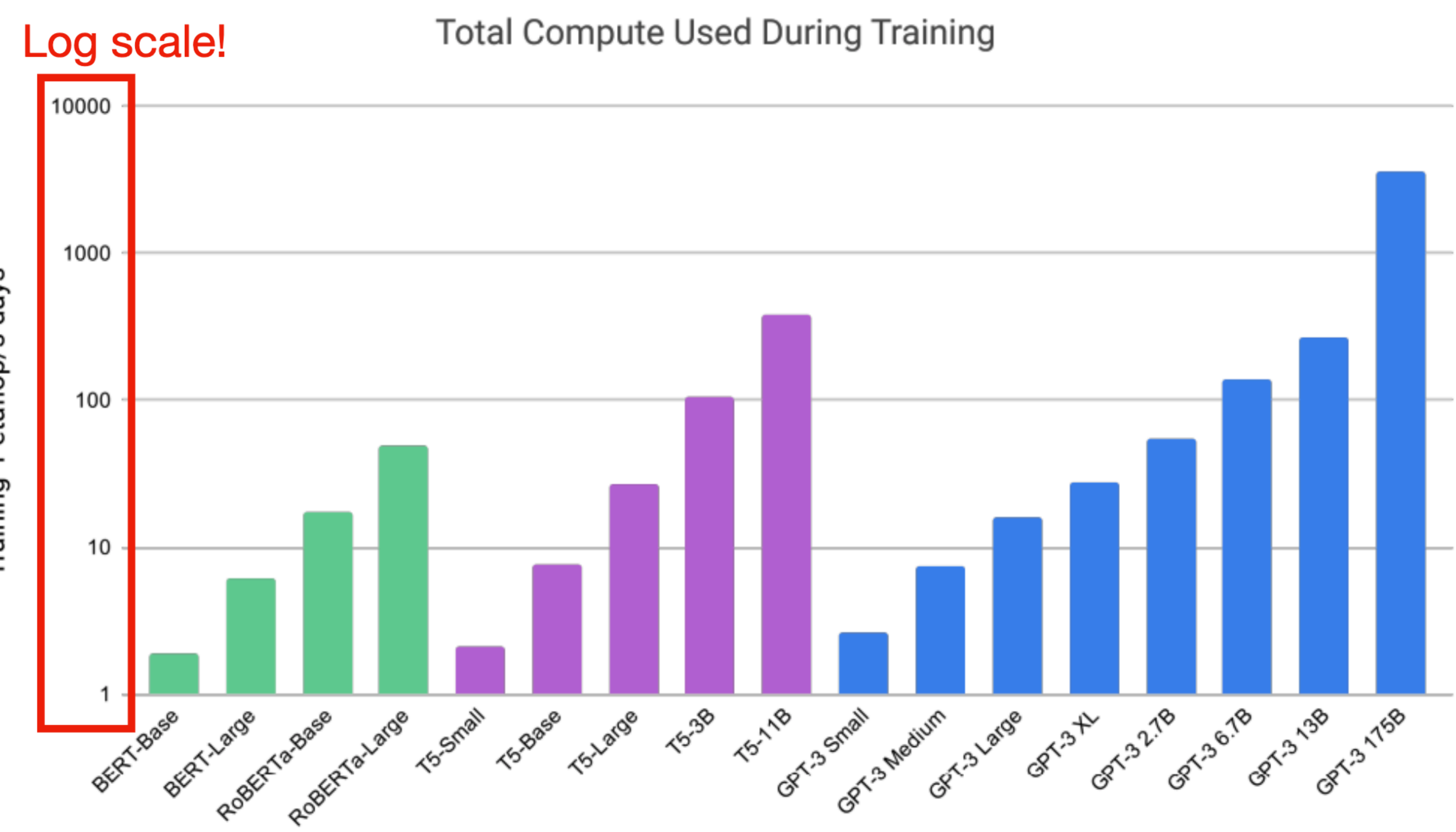
What if we tried to directly **learn** good text embeddings?

What Makes a Good Prompt?

+ Giving multiple examples
+ Specifying the answer format
+ Order matters!
+ Explanation seems key
+ Word puzzles seem harder than trivia

The Costs of Deep Learning

Models keep getting larger



Models keep getting larger

2022-2023:

PaLM (Google): 540B params, 118 layers, 18432 d_model, 780 billion training tokens **Model not available**

ChatGPT (OpenAI): Params, layers, dimensionality, training data size unknown **Model available only through blackbox API**

LLaMa (Meta): 65B params, 80 layers, 8192 d_model, 1.4 trillion tokens of training data **Model parameters publicly available!**

GPT4 (OpenAI): Params, layers, dimensionality, training data size unknown **Model available only through blackbox API**

Bard (Google): Params, layers, dimensionality, training data size unknown **Model available only through blackbox API**

These models are really expensive!

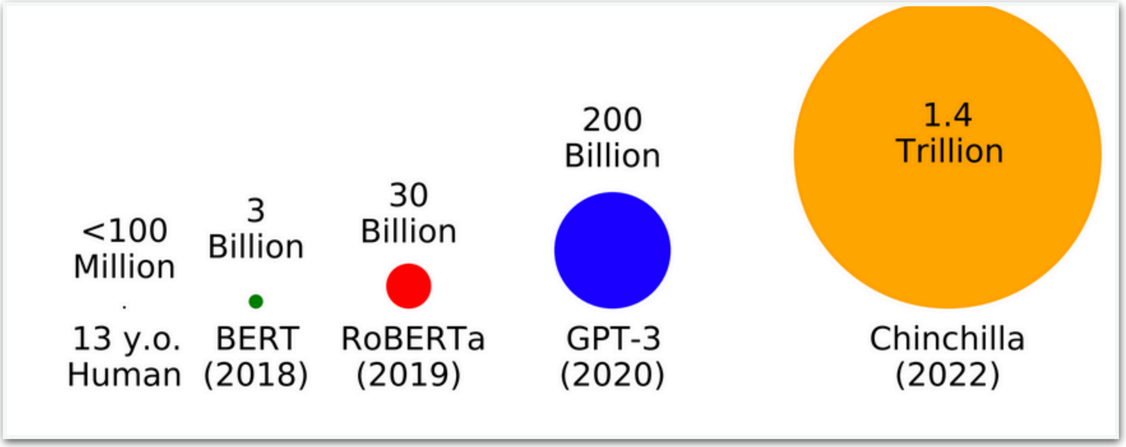
Megatron (530 billion parameters), Microsoft's GPT-3 competitor, cost around **$100 million** to train

These models are really expensive!

www.lesswrong.com/posts/midXmMb2Xg37F2Kgn/new-scaling-laws-for-large-language-models

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Pathways (Chowdhery et al. 2022)	540 Billion	780 Billion

By contrast: children are exposed to < 100 million words in their critical language acquisition period.



babylm.github.io

Baby LLM Project

Alex Warstadt is giving a talk about language acquisition in LLMs versus humans in my NLP class (9:55am) on **Nov. 28th**!

BabyLM Challenge

Sample-efficient pretraining on a developmentally plausible corpus

Overview • Guidelines • Timeline • FAQs

Summary: This shared task challenges community members to train a language model **from scratch** on the same amount of linguistic data available to a child. Submissions should be implemented in Huggingface's Transformers library and will be evaluated on a shared pipeline. This shared task is co-sponsored by CMCL and CoNLL.

- Download Dataset (700MB unzipped)
- Evaluate your model using our evaluation pipeline
- Models and results due ~~**July 15, 2023**~~ **July 22, 2023, 23:59 anywhere on earth (UTC-12)**. Submit on dynabench.
- Paper submission due ~~**August 1, 2023**~~ **August 2, 2023, 23:59 anywhere on earth (UTC-12)**. Submit on OpenReview.

See the guidelines for an overview of submission tracks and pretraining data. See the call for papers for a detailed description of the task setup and data.

Consider joining the BabyLM Slack if you have any questions for the organizers or want to connect with other

These models are really expensive!

Consumption	CO_2e (lbs)
Air travel, 1 person, NY↔SF	1984
Human life, avg, 1 year	11,023
American life, avg, 1 year	36,156
Car, avg incl. fuel, 1 lifetime	126,000
Training one model (GPU)	
NLP pipeline (parsing, SRL)	39
w/ tuning & experiments	78,468
Transformer (big)	192
w/ neural arch. search	626,155

Table 1: Estimated CO_2 emissions from training common NLP models, compared to familiar consumption.[1]

Strubell, Ganesh, & McCallum (2019)

Emma Strubell

These models are really expensive!

BERT-L (340 million parameters) had a **carbon footprint** equivalent to a trans-American flight.

And remember:

Microsoft Megatron has 530 **billion** parameters...

Google Pathways has 540 **billion** parameters...

Models keep getting larger