Large Language Models

Part I: Pre-training and Decoder-only Models

Aaron Mueller
CAS CS 505: Introduction to Natural Language Processing
Boston University
Spring 2026

Admin

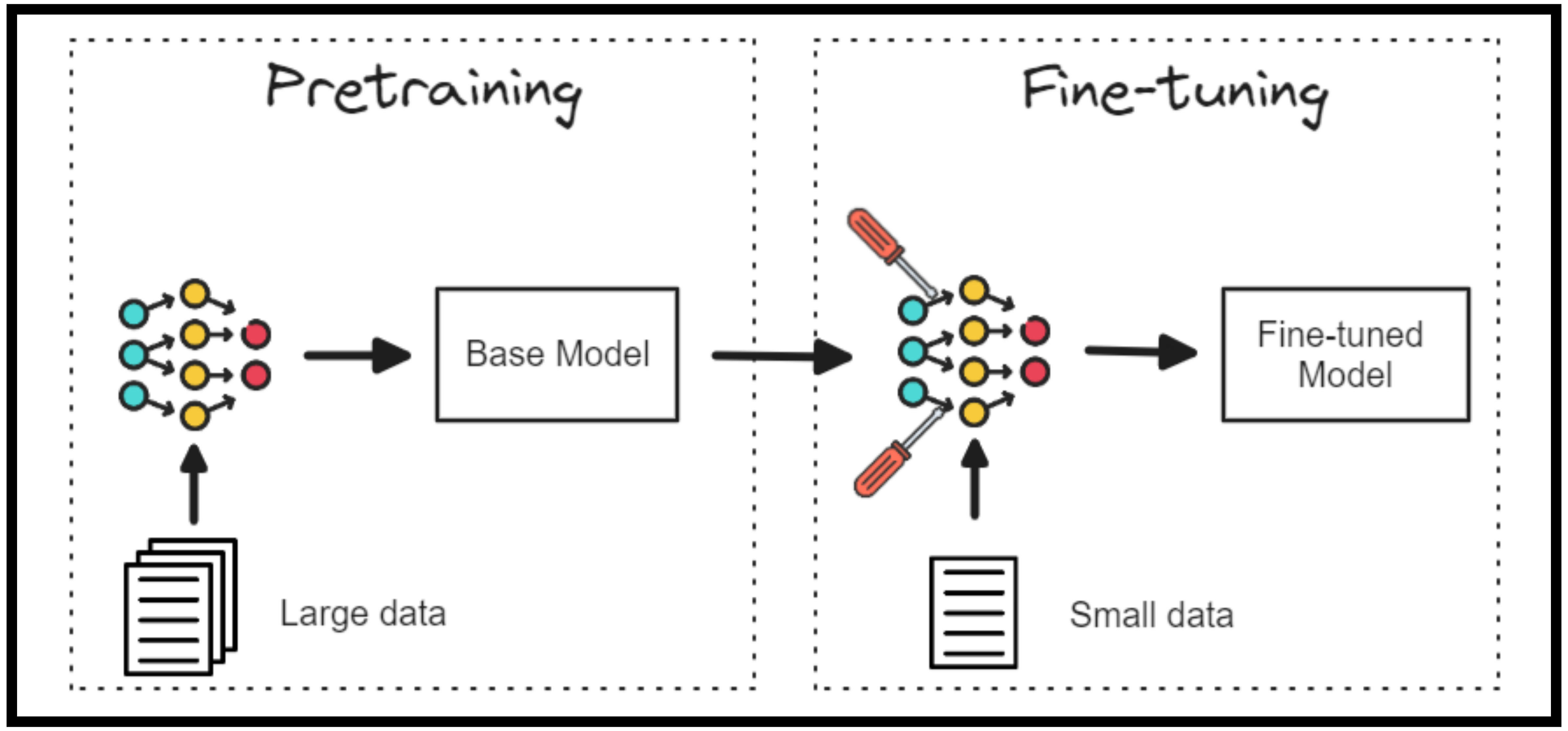
- **HW0** grades have been released on Gradescope
    - The average score was very high—congrats!

- **HW1** is due in one week: **Feb. 19**, at 11:59pm
    - We'll have a homework help session during your labs on *Tue., Feb. 17*. This is a Monday schedule because of the holiday

- **HW2** has been released! This will be due in *three weeks*, on **Mar. 5**

Overview of Concepts

Pretraining is the way we optimize large Transformer-based models to be good general-purpose systems.



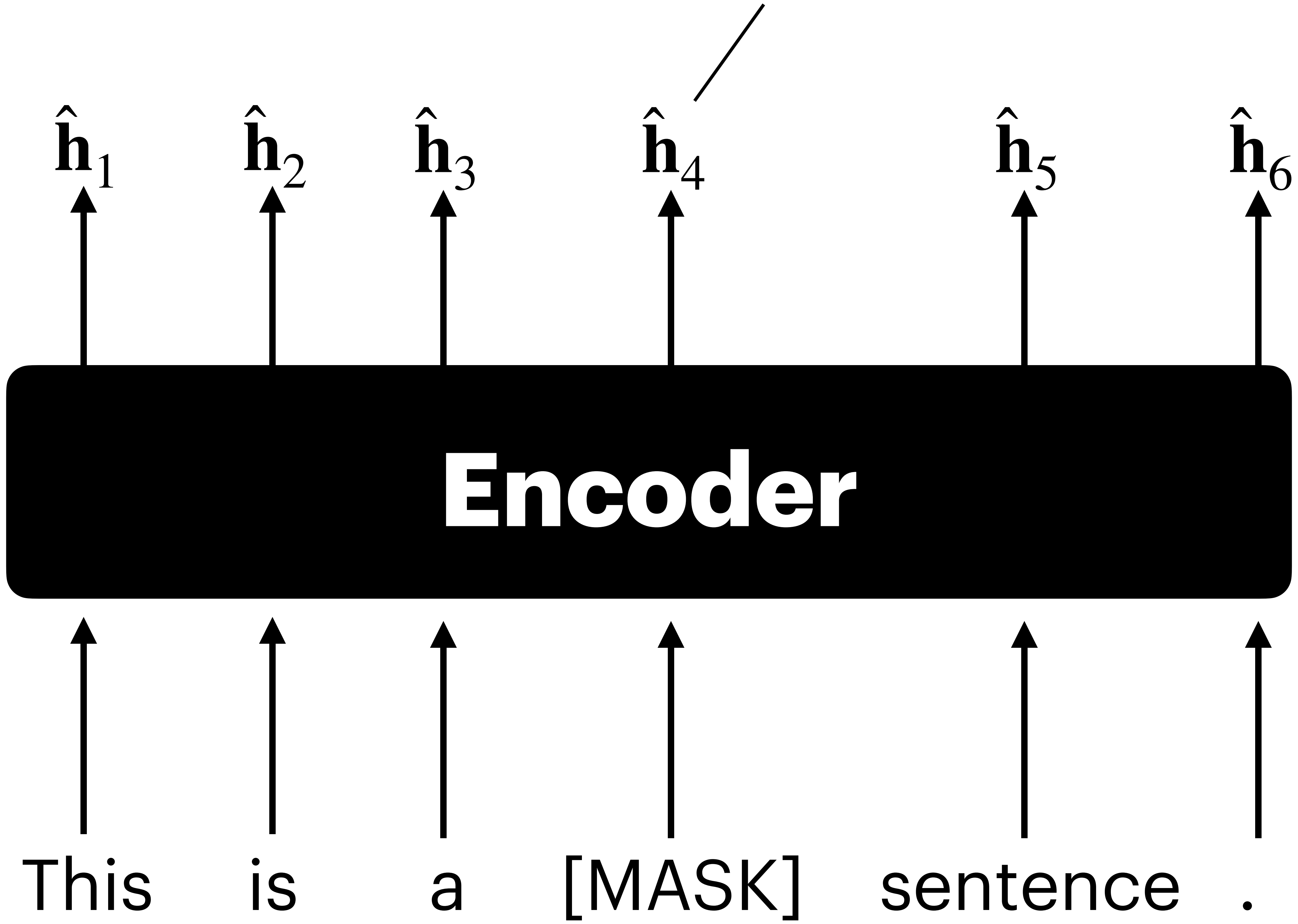
Decoder-only models are the most common type of Transformer-based model today.

Nucleus sampling allows us to generate more interesting *and* coherent outputs with LMs compared to other decoding methods.

Scaling laws describe the predictable improvements in performance we get from increasing dataset and model sizes.

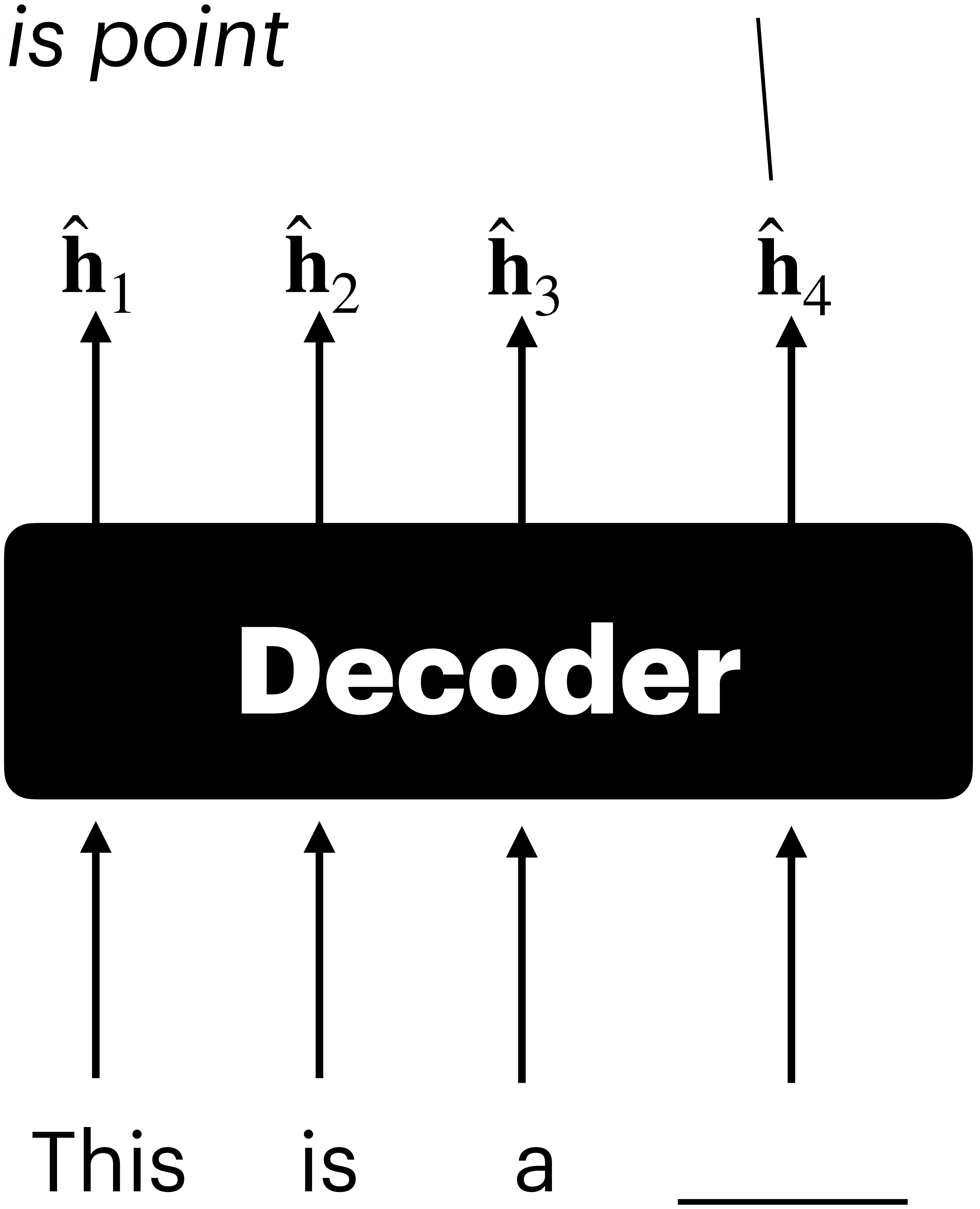
Encoder-only vs. Decoder-only Models

Uses left and right context, yields representation of all tokens



$\hat{\mathbf{h}}_1$ $\hat{\mathbf{h}}_2$ $\hat{\mathbf{h}}_3$ $\hat{\mathbf{h}}_4$ $\hat{\mathbf{h}}_5$ $\hat{\mathbf{h}}_6$
Encoder
This is a [MASK] sentence .

Model sees everything up to this token, yields representation of all tokens up to this point



$\hat{\mathbf{h}}_1$ $\hat{\mathbf{h}}_2$ $\hat{\mathbf{h}}_3$ $\hat{\mathbf{h}}_4$
Decoder
This is a ______

Don’t pay too much attention to this terminology:

Both models “encode” tokens—but decoders only encode up to the current position.

Encoders can technically “decode” one token—but decoders can generate far more.

Language Modeling

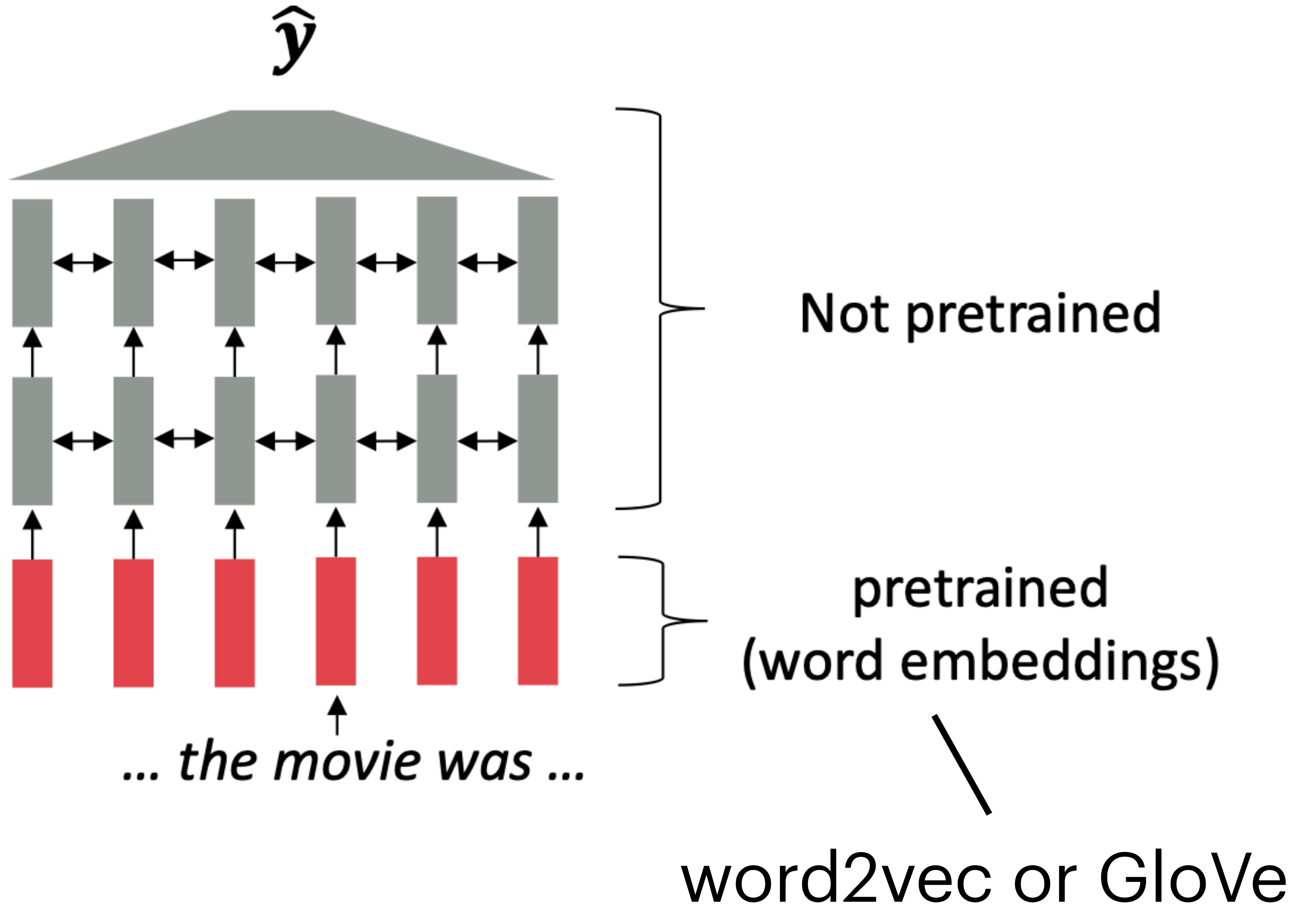
Then

- We could download pretrained word embeddings, but these embeddings do not depend on context.
- Imagine using the same representation for all of these occurrences of “park”:

  Let’s **park** the car.

  I went to the **park**.

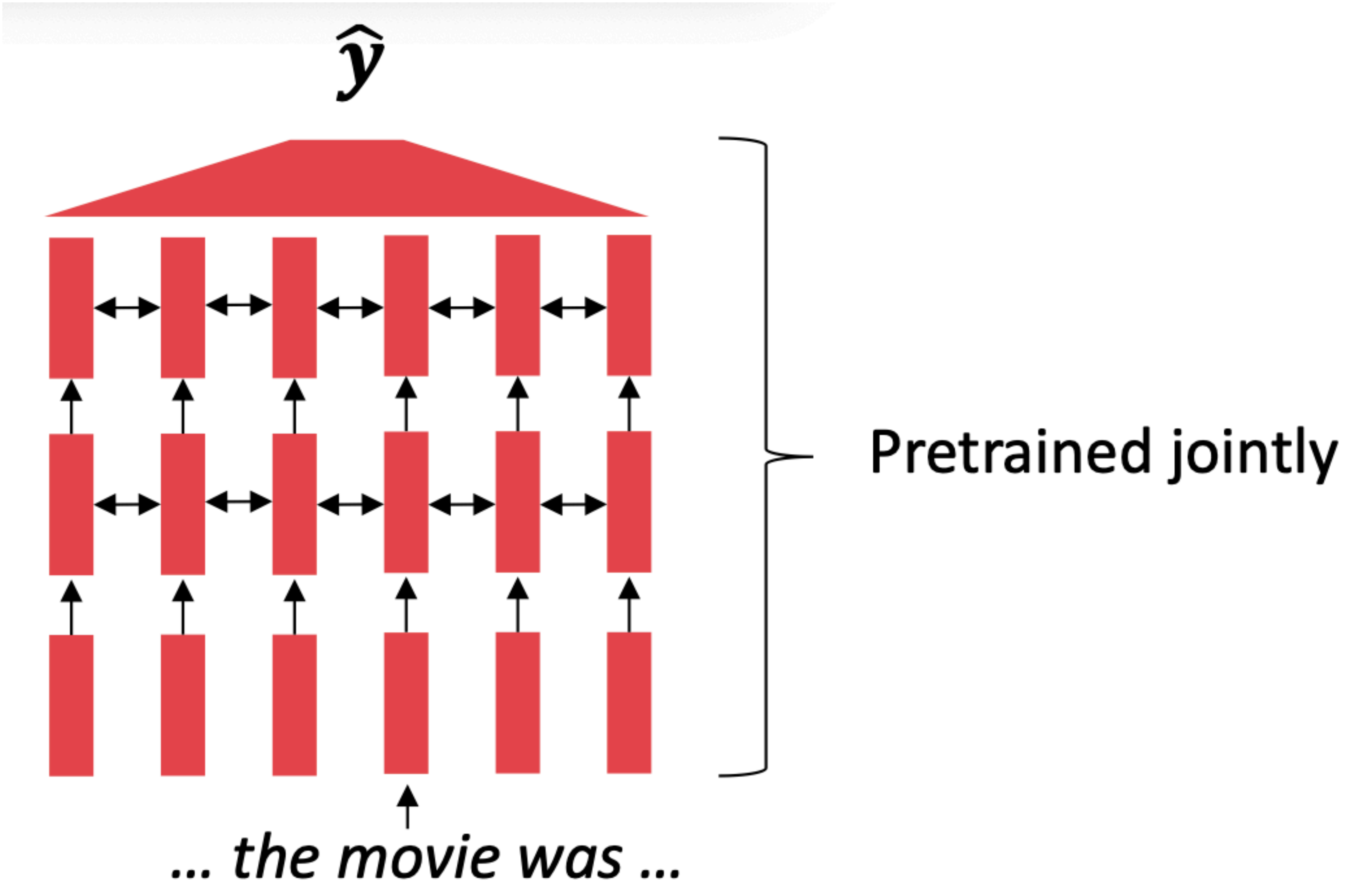
- Most parameters must be learned. Thus, your training data must be sufficient to teach the model all important aspects of language.



Language Modeling

Now

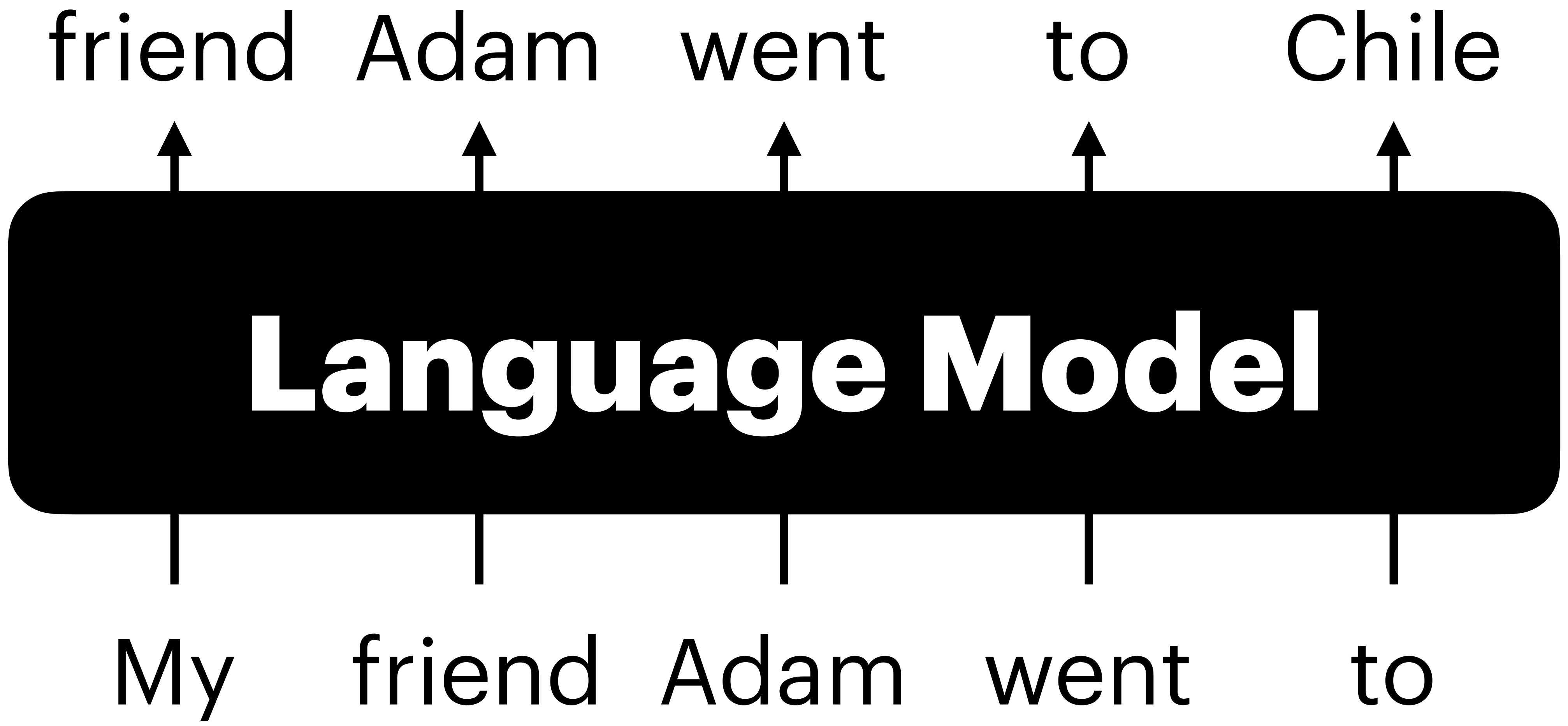
- These days, everything is **jointly** learned during pretraining on *very large* datasets—including embeddings. This has been *enormously* effective.
- Language modeling used to be a downstream task: people did it to study aspects of language and its distribution.
- Nowadays, we usually think of it as the first step in a two-step pipeline: (1) pretrain via language modeling and then (2) fine-tune for a specific task.



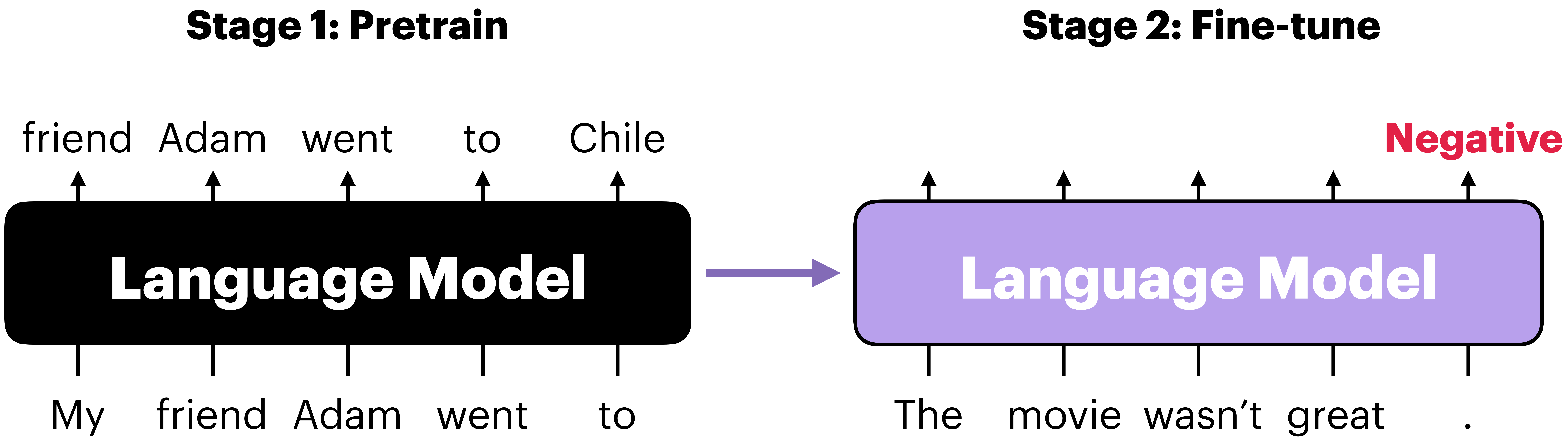
Why pretrain with the language modeling task?

- Collecting data for language modeling is very easy.
  - (For English)

- We can give a model a lot of language modeling data so that it can learn embeddings and all of its weights.
- We save its parameters, load them later, and adapt them for a downstream task where data is more scarce.



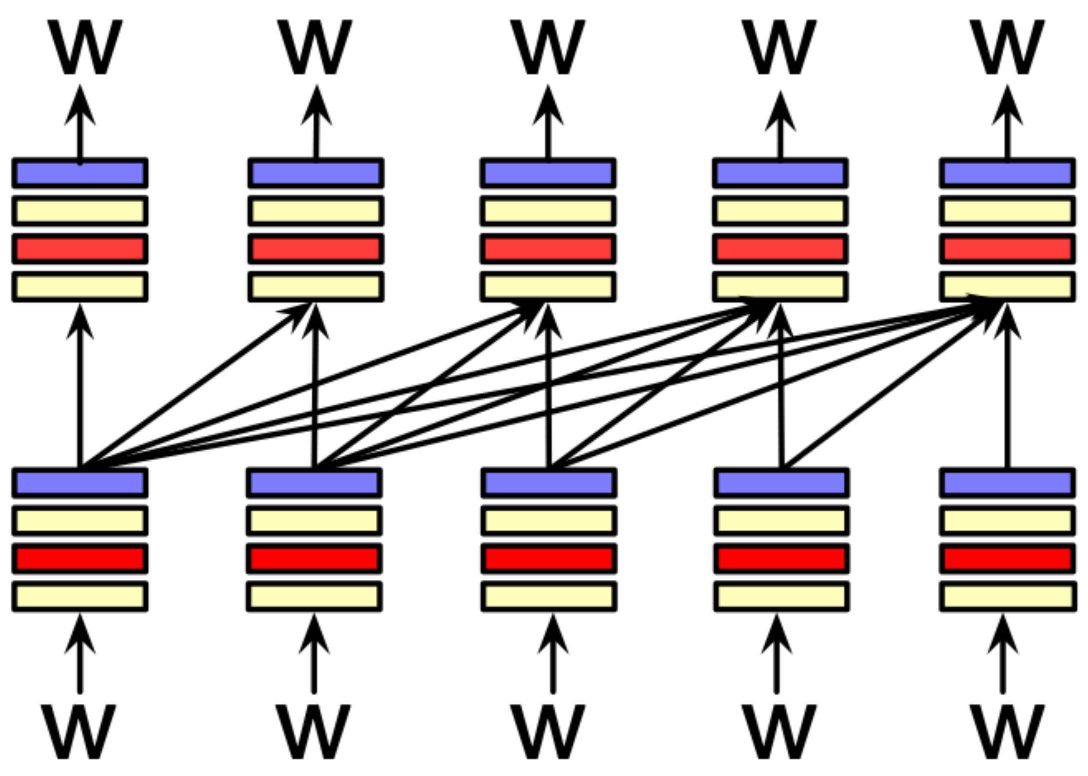
Transfer Learning



Why should this work at all?

Transfer learning: pretraining teaches the model general aspects of language, like word meanings, factual information, world models, etc. This knowledge *transfers* well to other domains and tasks.

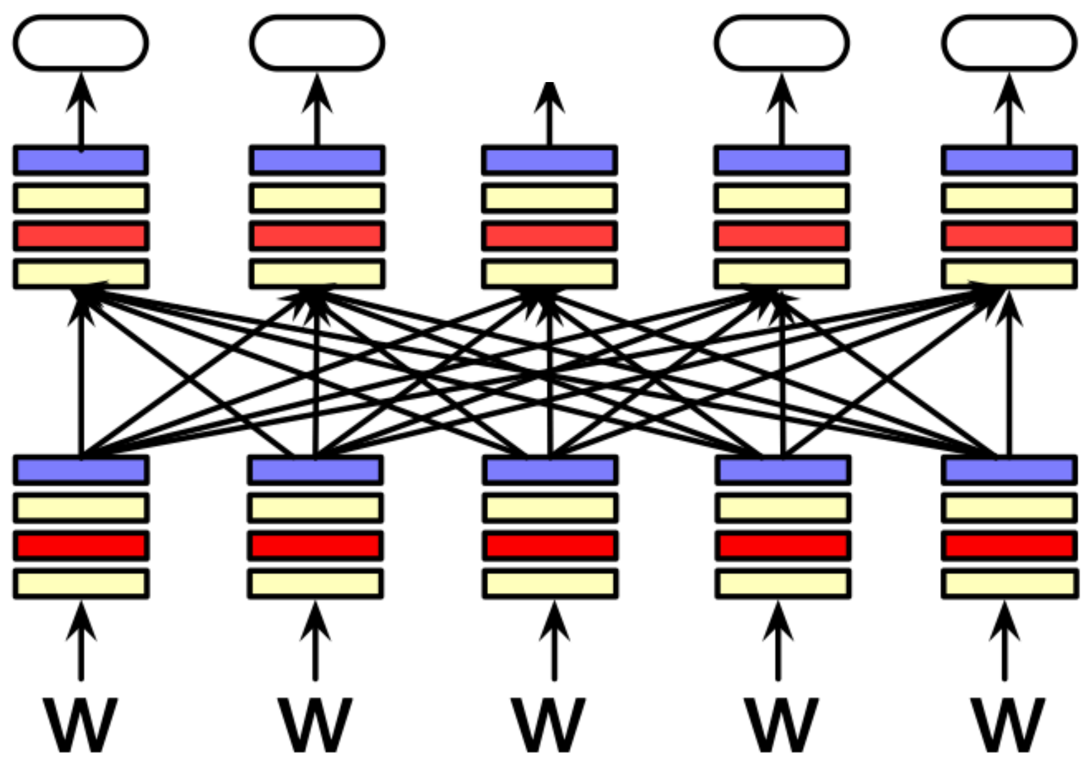
Architectures



Decoder

Left-to-right: generate the next word given prior context.
Language models!

Good for generation.

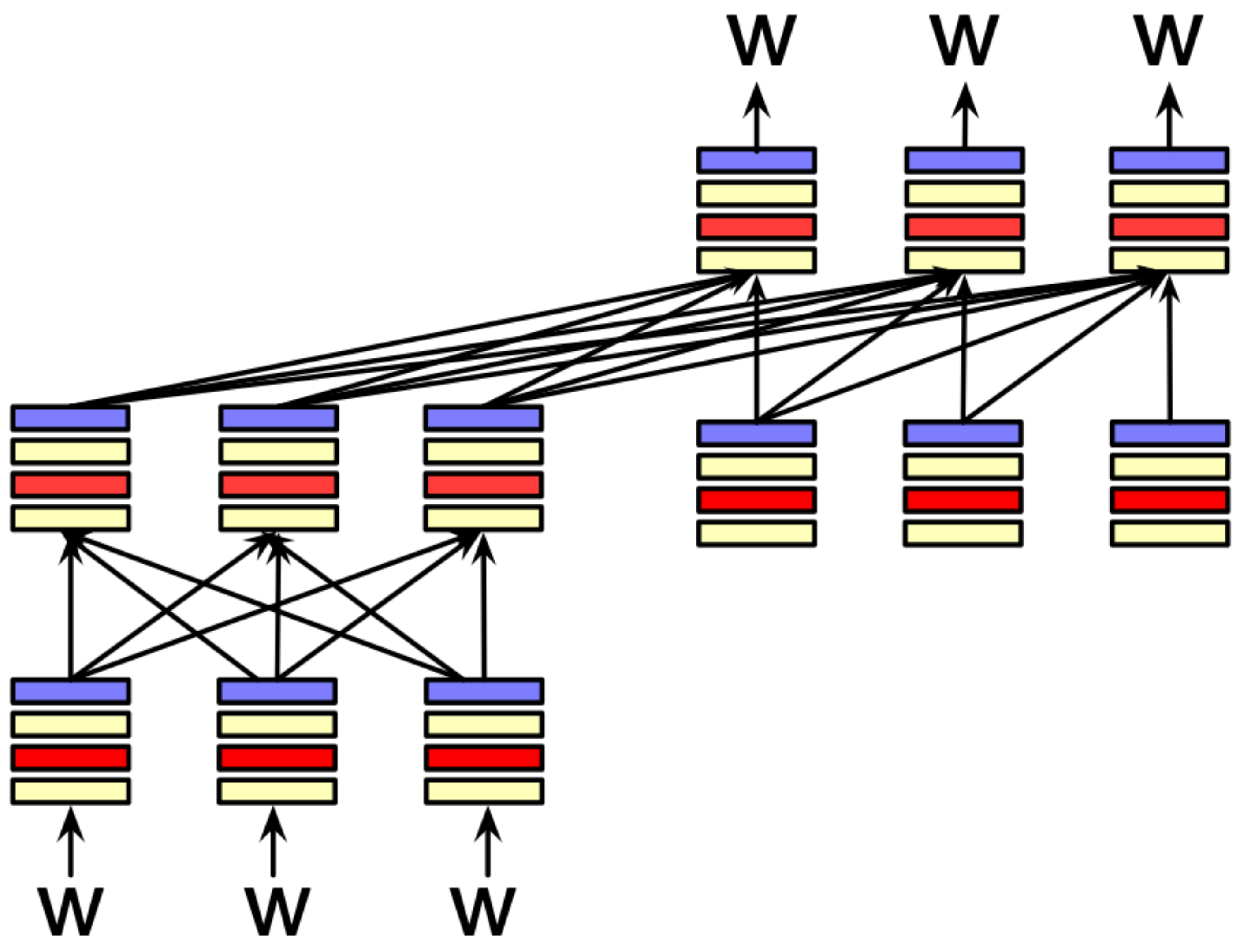


Encoder

Bidirectional: predict a word given left *and* right context.

Can only generate 1 token.

Good for classification.

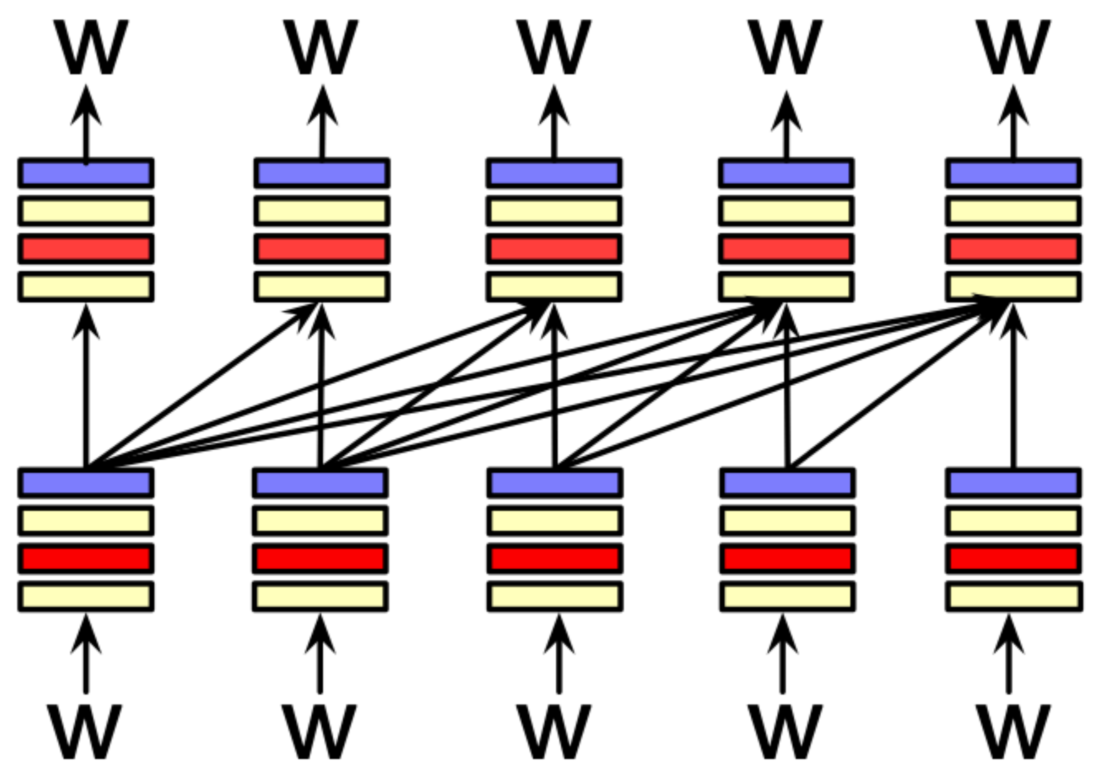


Encoder-Decoder

Best of both worlds, maybe?

Used to be more common; not so much anymore.

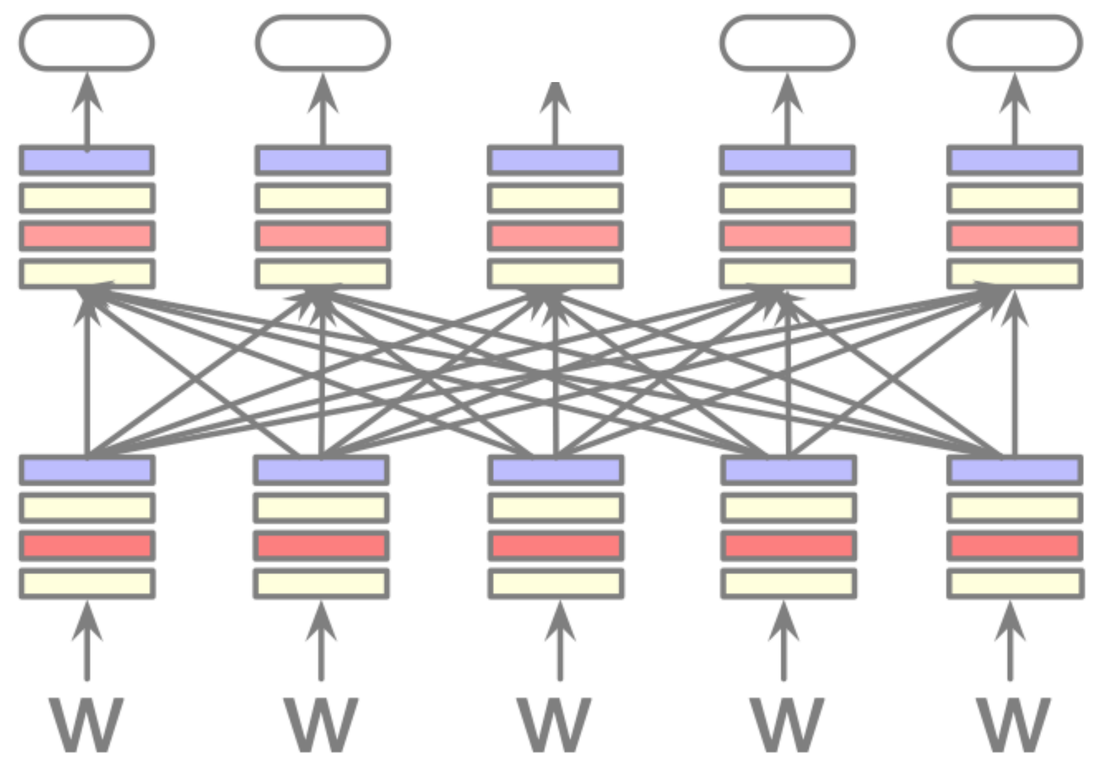
Architectures



Decoder

Left-to-right: generate the next word given prior context. Language models!

Good for generation.

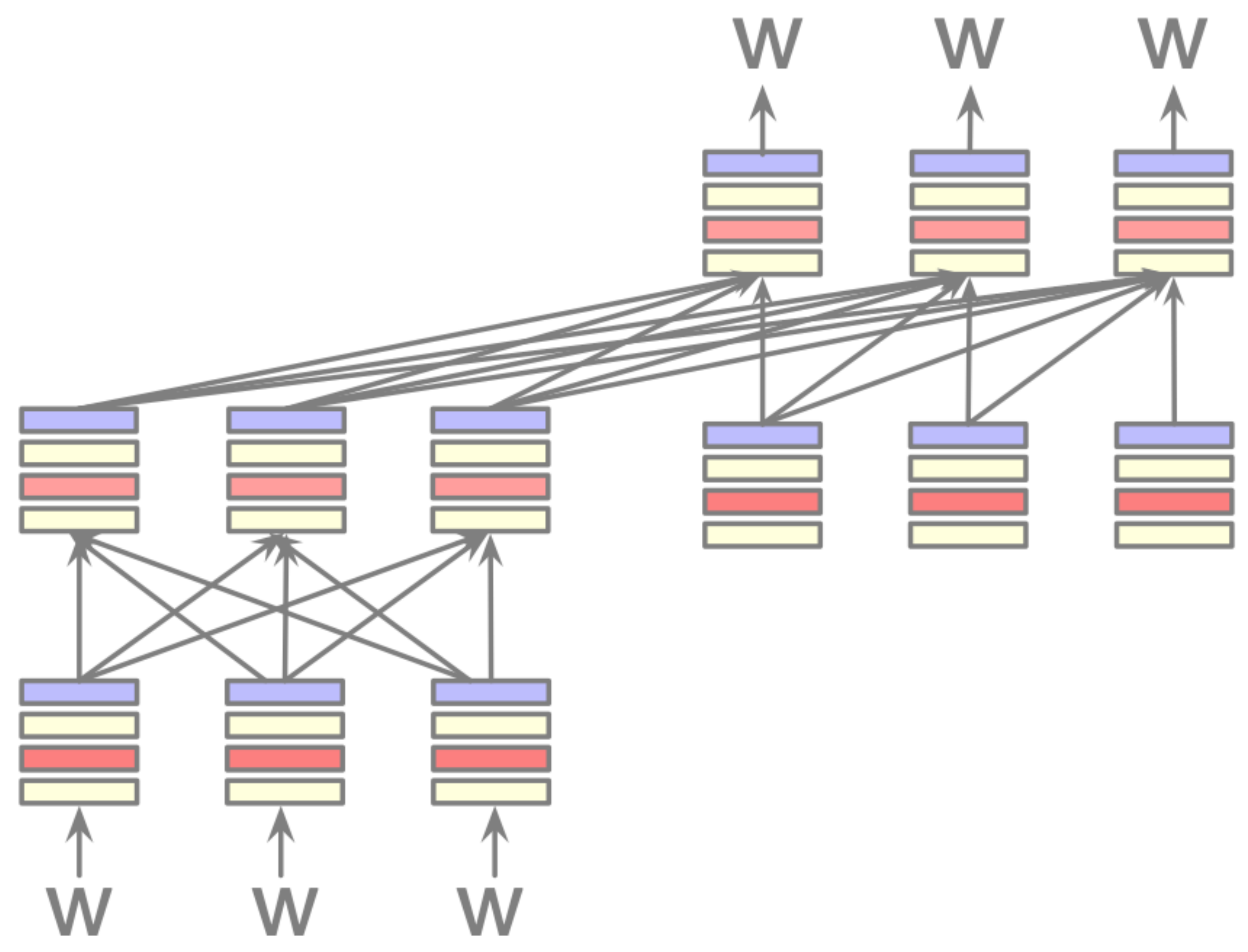


Encoder

Bidirectional: predict a word given left *and* right context.

Can only generate 1 token.

Good for classification.



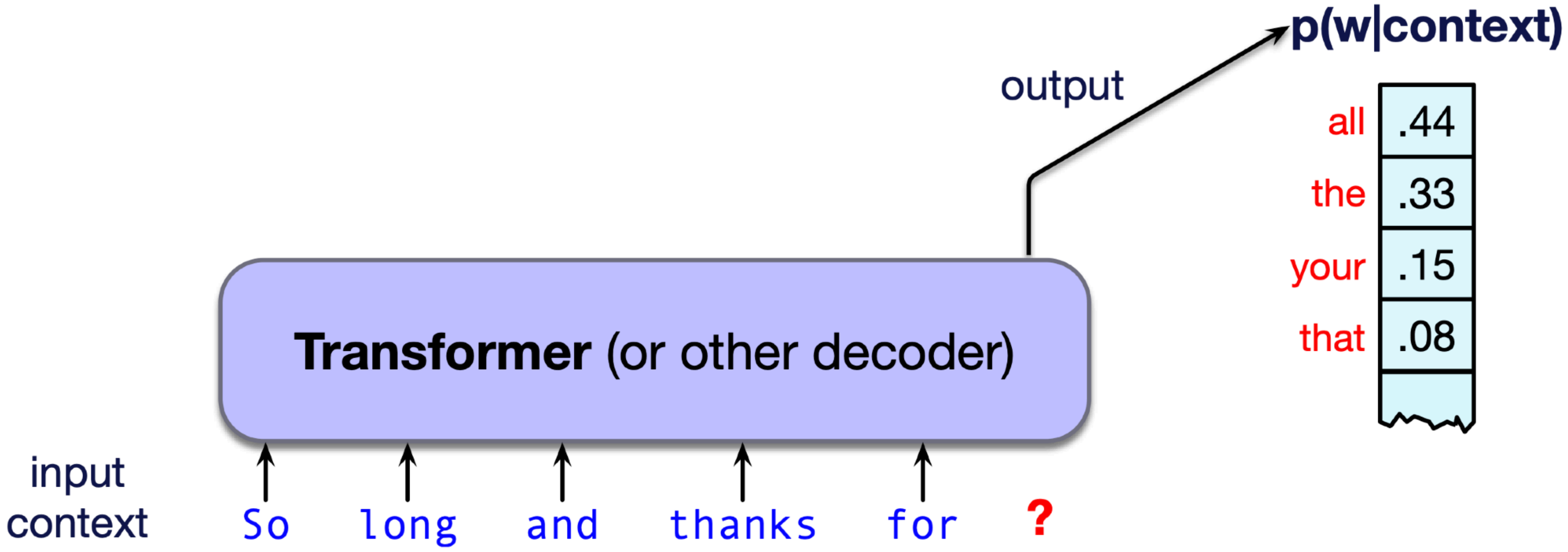
Encoder-Decoder

Best of both worlds, maybe?

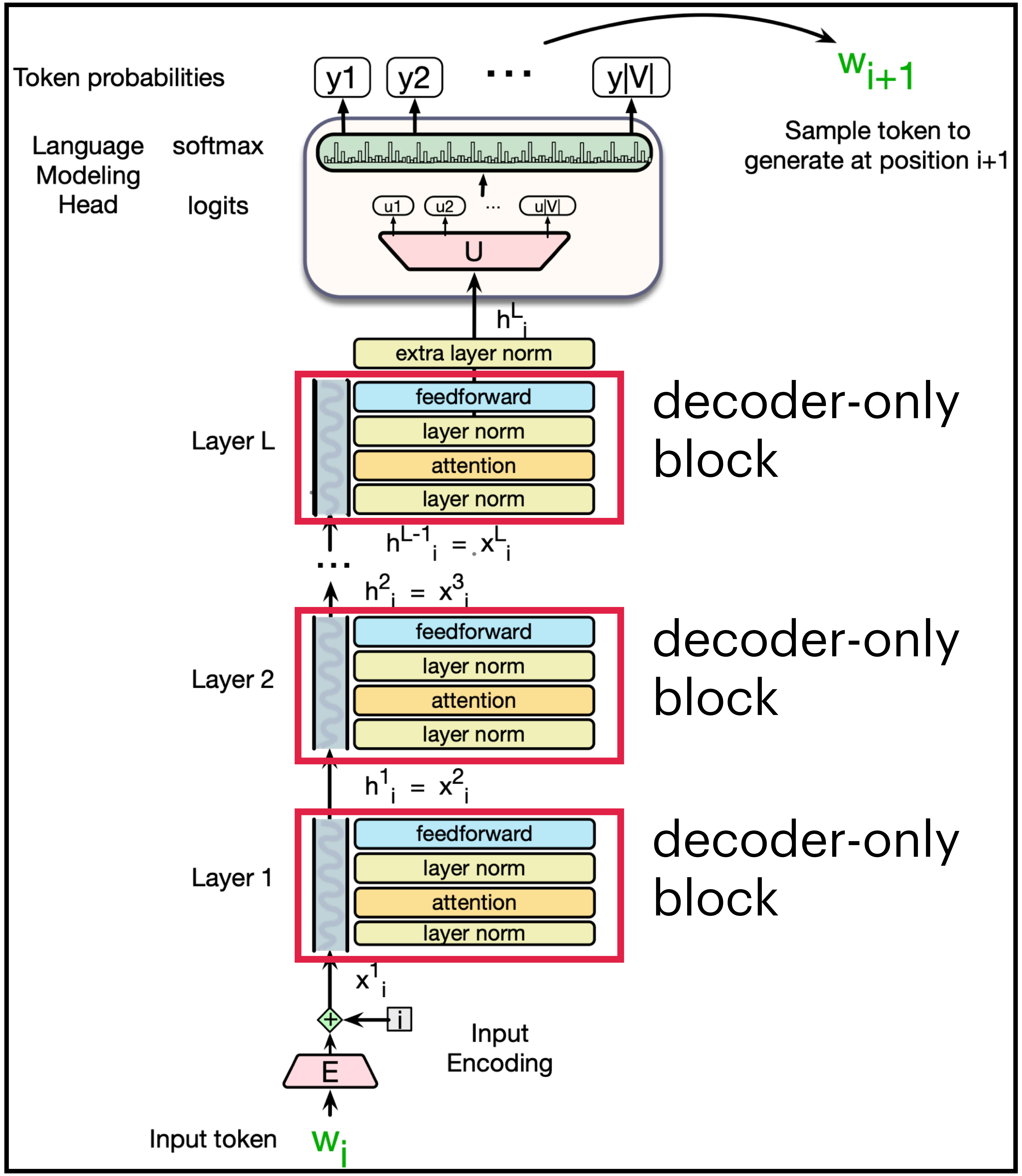
Used to be more common; not so much anymore.

Decoder LMs

- Decoder LMs (or decoder-only models) include GPT, Claude, Llama, Mistral, DeepSeek, and other famous LLMs.
- Idea: almost anything we want to do with language can be modeled as **conditional generation** of text.
    - We give an LLM some input text (the **prompt**), and have the LLM continue generating token-by-token, conditioned on the prompt and generated tokens.



- We can have multiple decoder layers one after the other.
- Important hyperparameters:
    - *Number of layers*
    - Number of attention heads
    - Hidden dimensionality
    - Feed-forward size
    - Vocabulary size
    - Context length
- If you have a certain parameter budget, it is usually best to scale the number of layers first.

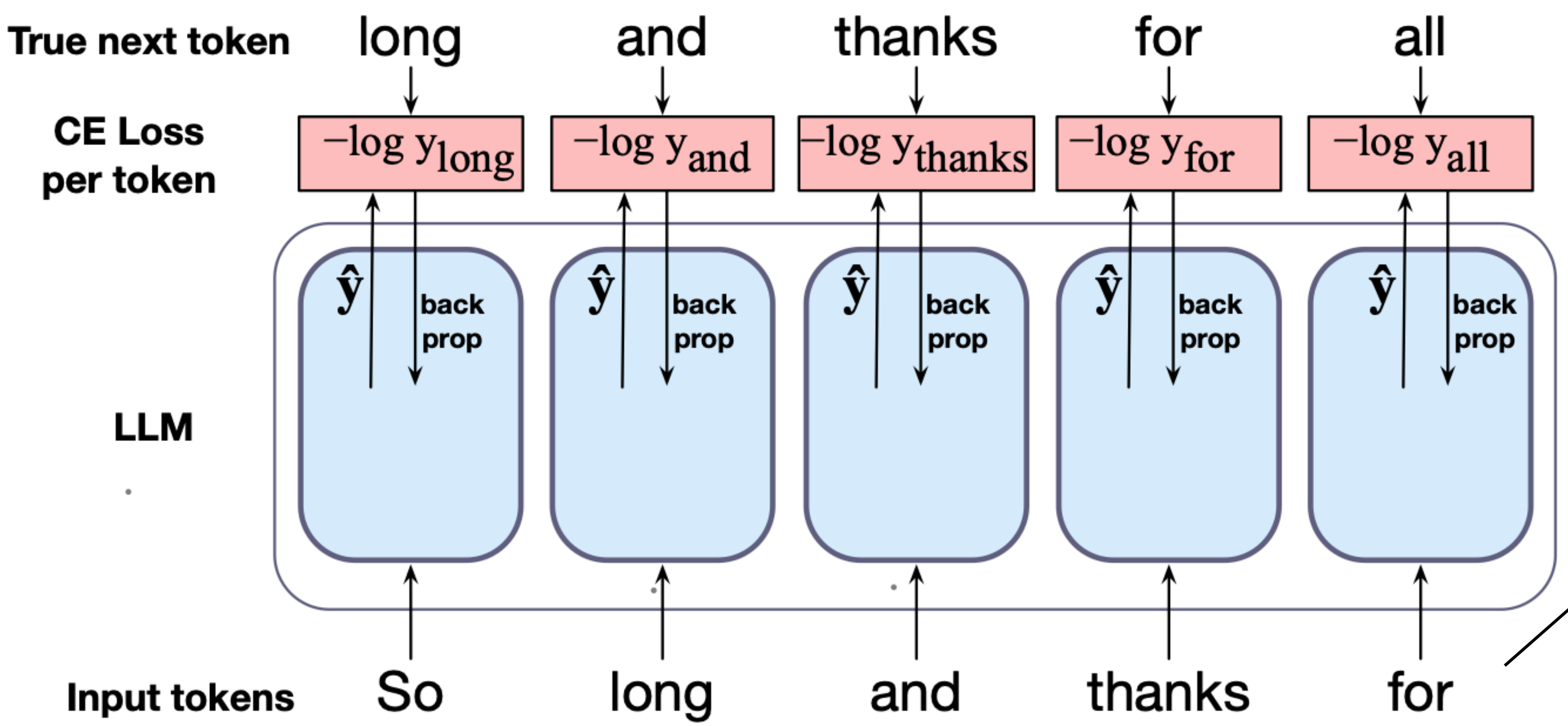


Example: GPT-2

- Decoder LLMs are trained using the same cross-entropy loss we’ve used so far.
- 2018: GPT
- 2019: GPT-2 (We’ll focus on GPT-2 small.)
    - Decoder-only model with 12 layers, 12 attention heads, 768-dimensional hidden states, 3072-dimensional feed-forward size, context width 1024
    - BPE tokenizer with vocab size ≈50,000
    - Trained on WebText (>8 million web pages)

Pretraining

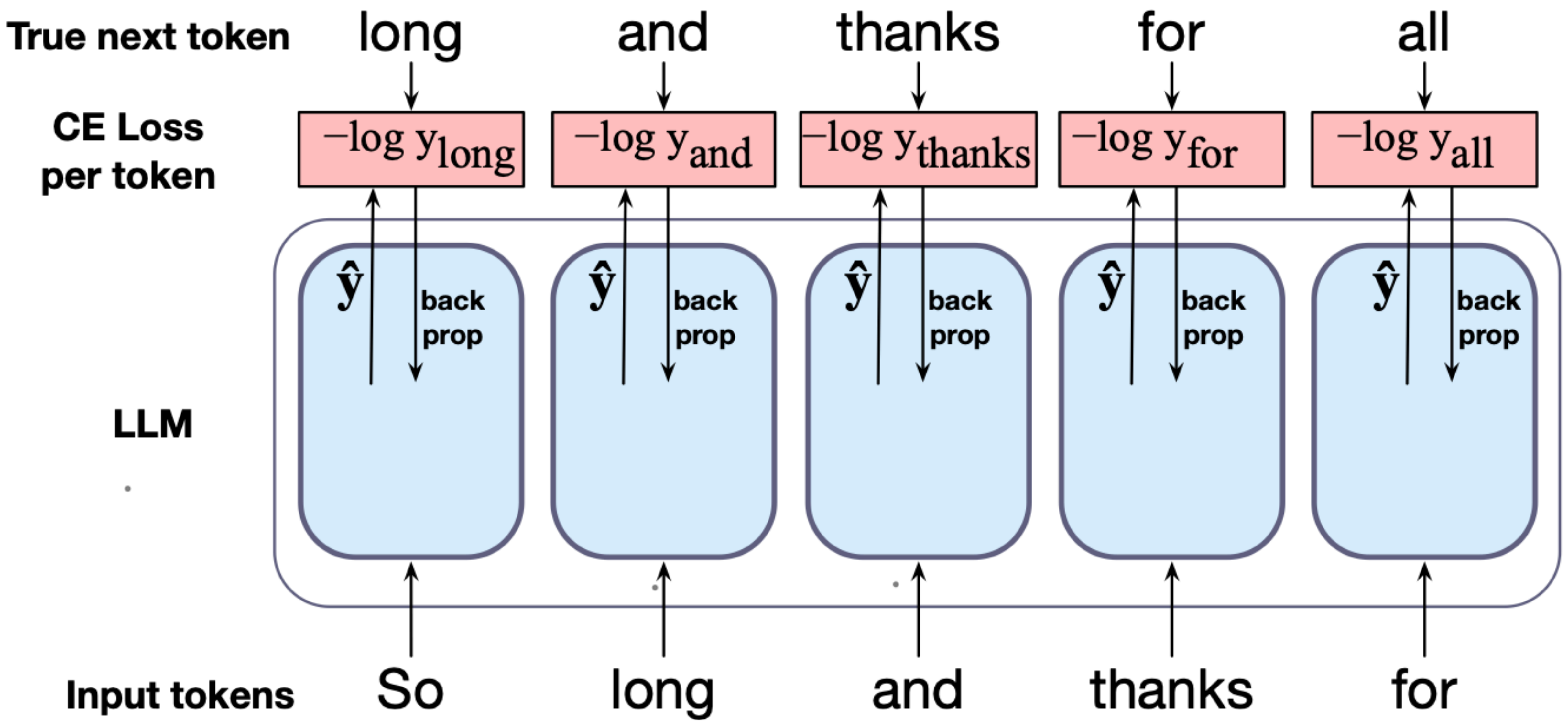
- The intuition is the same as for RNNs: **self-supervised** learning of token representations (and all other parameters in the model).
- Cross-entropy loss: $$L_{\text{CE}}(\text{batch}) = \frac{1}{T}\sum_{t=1}^{T} -\log \hat{\mathbf{y}}_t[w_{t+1}]$$



We use **teacher forcing** like before: regardless of what the LM generated, we always condition on the *true* context.

Pretraining

- All network weights are adjusted to minimize the average cross-entropy loss across the batch of tokens via gradient descent.
  - *All* weights: embedding matrix, query/key/value matrices, feed-forward layers, LayerNorm parameters, etc.

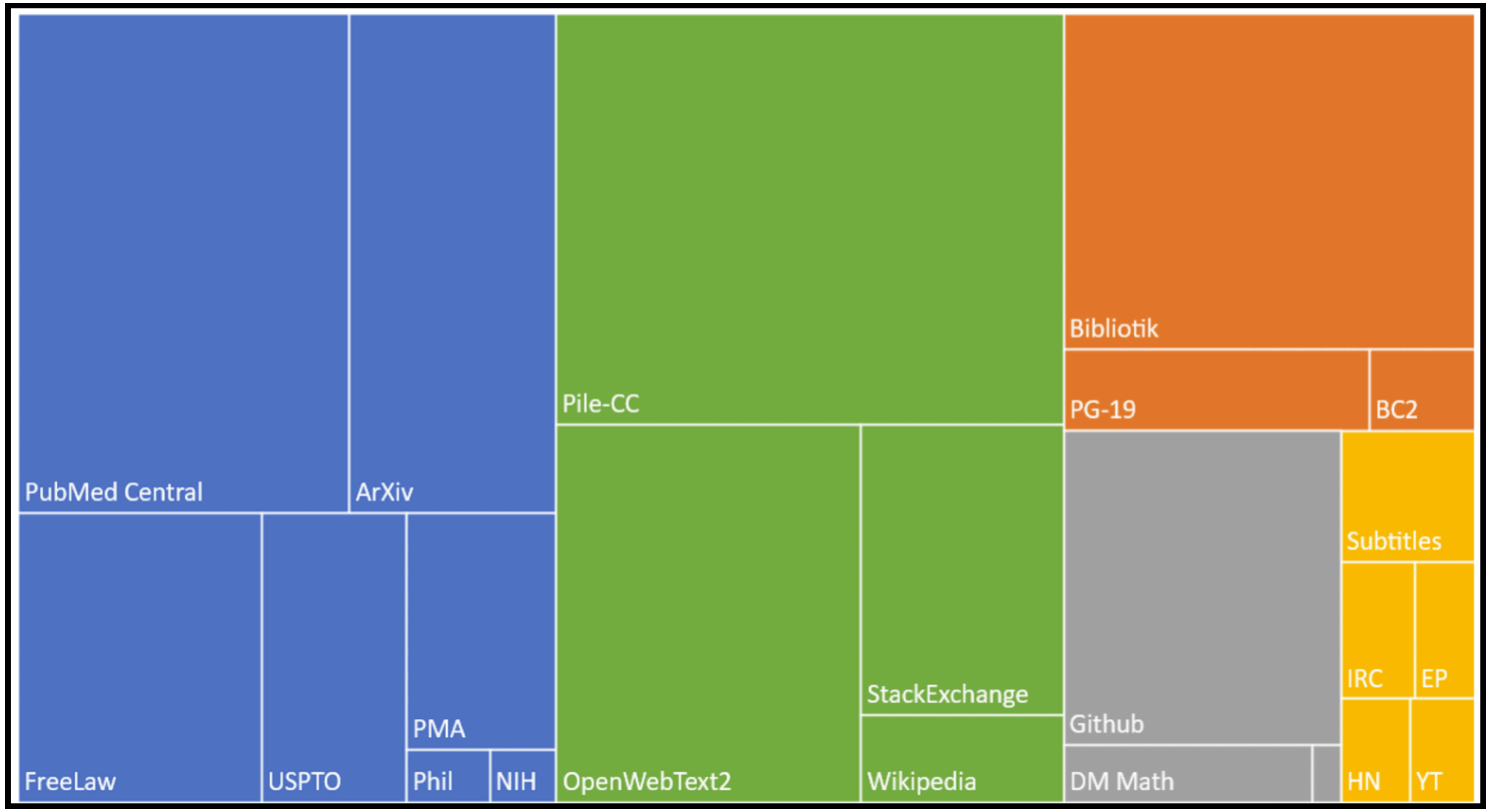


Pretraining Data

Where should our training data come from? *The more the better.*

- Internet webpages: CommonCrawl, the Colossal Clean Crawled Corpus (C4)
- Books: BookCorpus
- Combinations of sources: **The Pile**

Contains *academic*, *internet*, *prose*, *dialogue*, *miscellaneous* sources



Pretraining Data

Quality and Safety Considerations

The internet is not known as a bastion of kind and considerate content.

Pretraining data is usually filtered for **quality** and **safety**: we don’t want garbage text, nor text that is harmful or toxic.

There are also ethical and legal questions in using internet text:

- Copyright: not yet clear what the state of copyright is for data used to train AI systems
- Data consent: owners of websites may not want webcrawlers on their sites
- Privacy: Some data could include peoples’ phone numbers, emails, or other personally identifiable information (PII)
- Skewed distributions: most internet content comes from English-speaking Americans

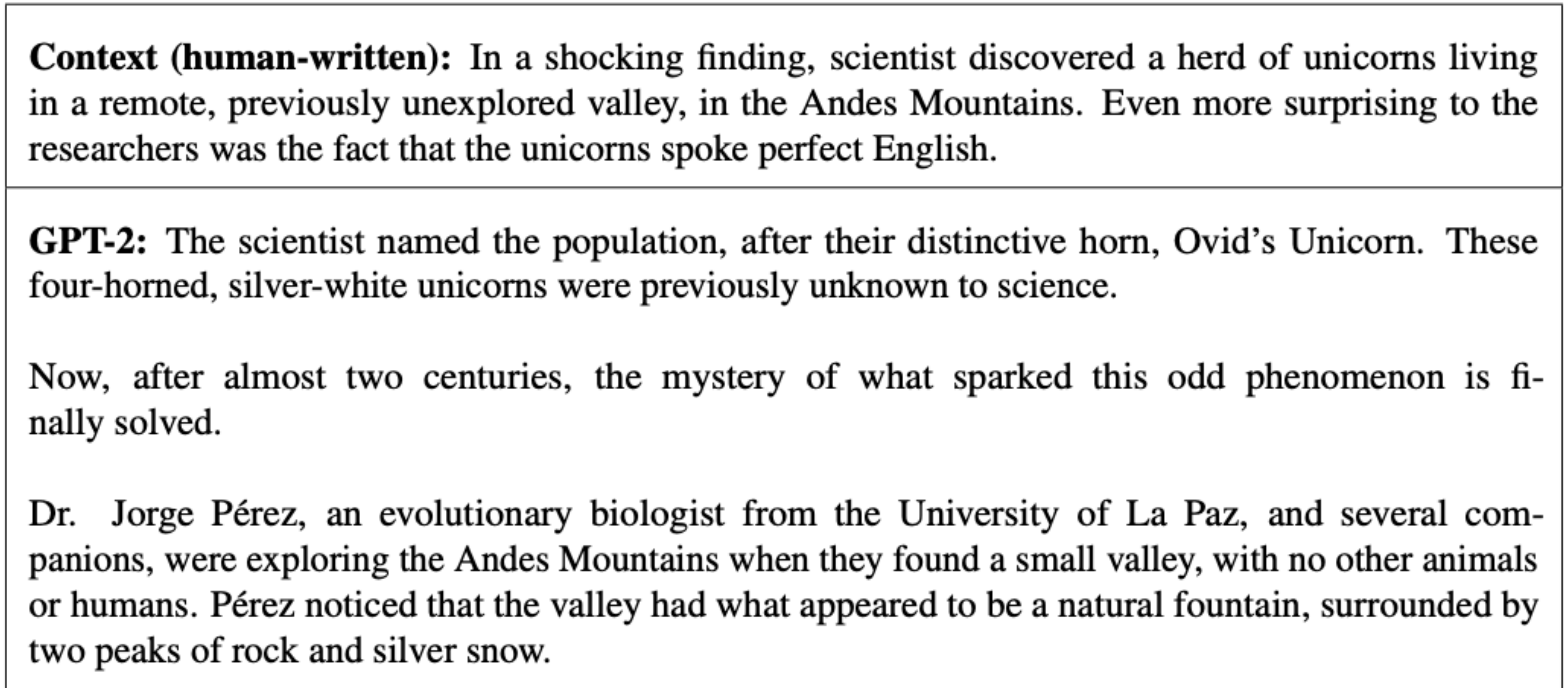
LSTM trained on Wikipedia **[Karpathy, 2015]**:

Branding
About 200,000 to about 21 million islands of [[Algeria]] and [[Spain]], account from Andorra. However, Durham was said to be the 2001 August [[12th century]] eastern New Gulhon, [[Guatemala State|Great Britain]], now claimed to have been diseduted by accession and tomato, for lands an operation of [[New Spring Party|New Zealand]] province annexed Curly Telegraph decades of parent prisoners. By 1925, Finland has earned the right in Czech, and lasting the supremacy and therefore.

This text doesn't make much sense.

GPT-2 produces surprisingly convincing natural language.

[Radford et al., 2015]

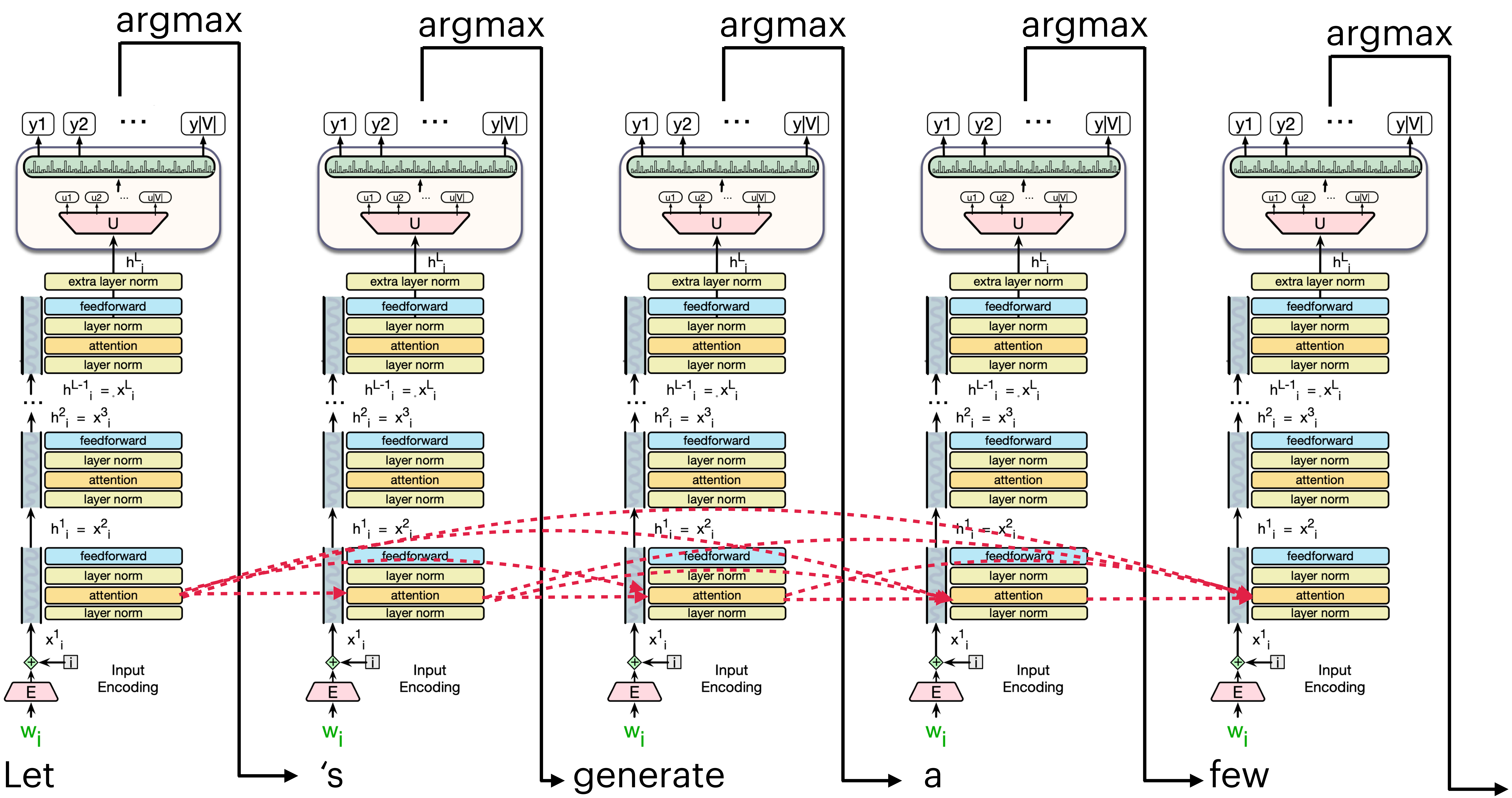


Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.


argmax
y1
y2
...
y|V|
u1
u2
...
u|V|
U
h^L_i
extra layer norm
feedforward
layer norm
attention
layer norm
$h^{L-1}_i = x^L_i$
...
$h^2_i = x^3_i$
feedforward
layer norm
attention
layer norm
$h^1_i = x^2_i$
feedforward
layer norm
attention
layer norm
x^1_i
i
E
Input Encoding
w_i
Let
'S
generate
a
few

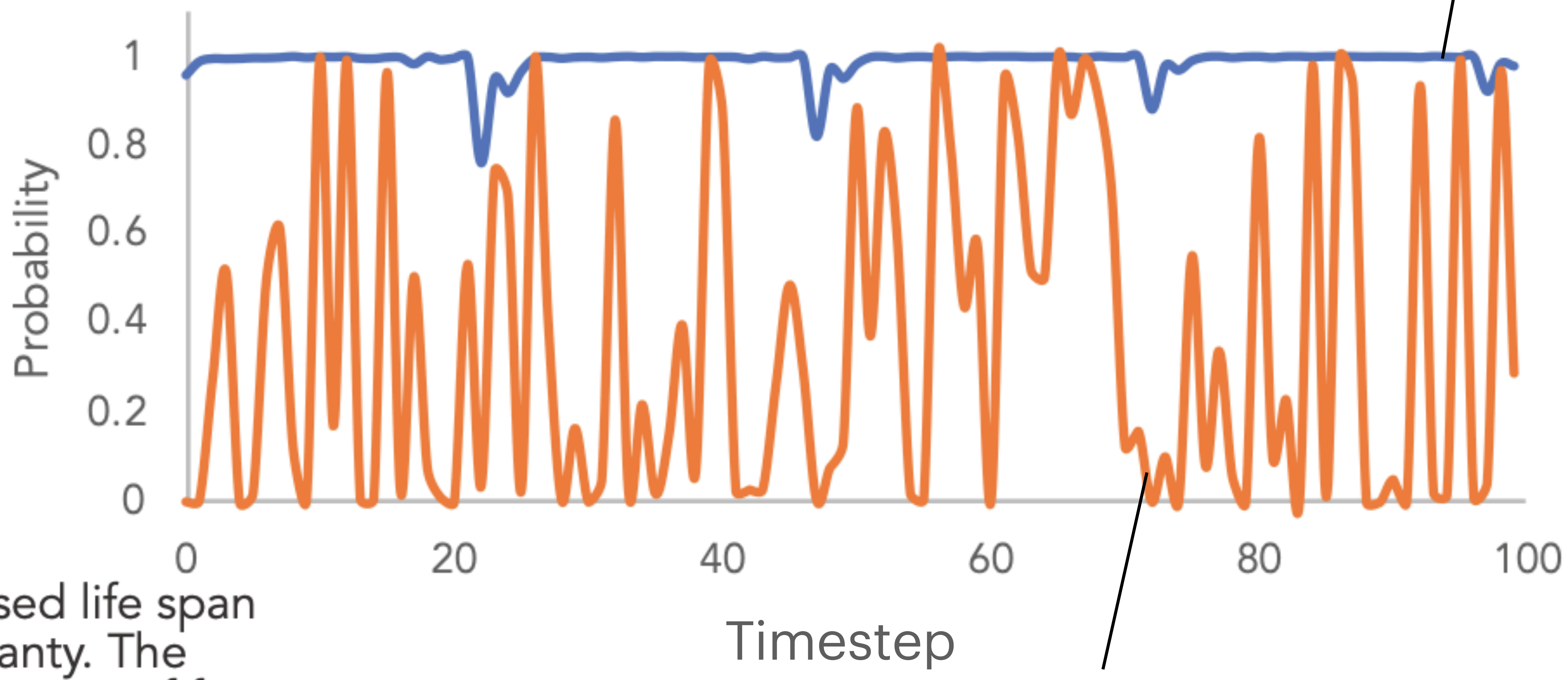
Generating with Decoders

- The previous example showed **greedy decoding,** where we just pick the most probable token at every timestep:

$$\hat{w}_t = \arg\max_{w \in V} p(w \mid \mathbf{w}_{<t})$$

- This is not the best generation method if you want interesting or creative outputs. *People tend to say surprising things!*
    - In fact, this is a formal requirement: the *Gricean maxim of quantity*. People generally optimize for their conversational contributions to be informative to the other person—but not too much nor too little

Human language is surprising.



Generated with an LLM

Human-written text

LM

...to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and to provide an overview of the current state-of-the-art in the field of computer vision and machine learning, and...

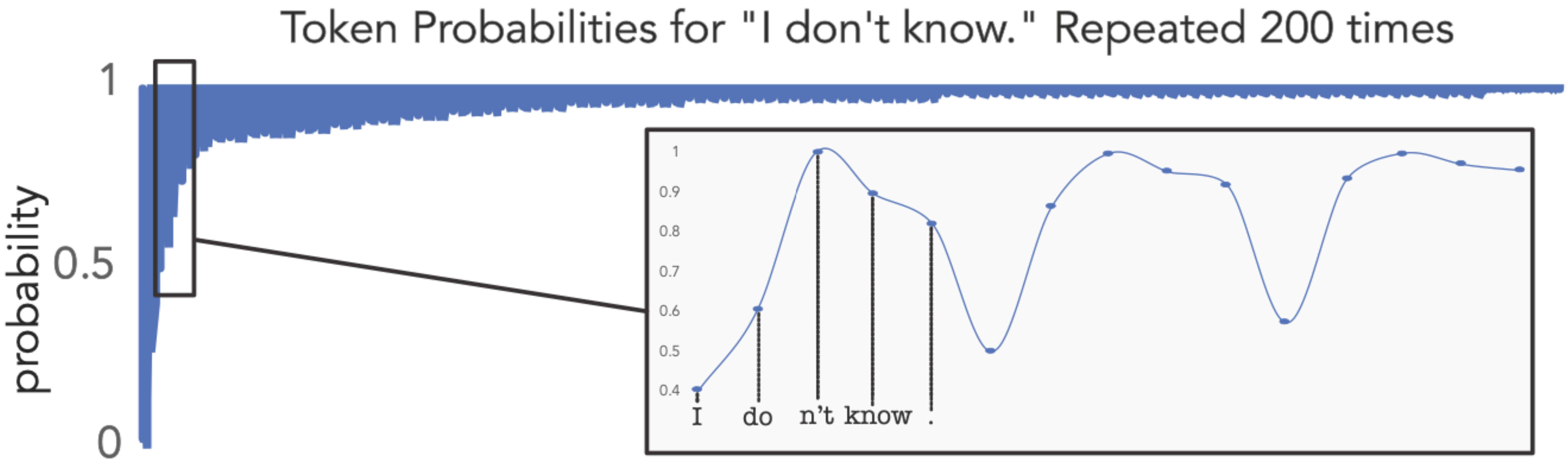
Human

...which grant increased life span and three years warranty. The Antec HCG series consists of five models with capacities spanning from 400W to 900W. Here we should note that we have already tested the HCG-620 in a previous review and were quite satisfied With its performance. In today's review we will rigorously test the Antec HCG-520, which as its model number implies, has 520W capacity and contrary to Antec's strong beliefs in multi-rail PSUs is equipped...

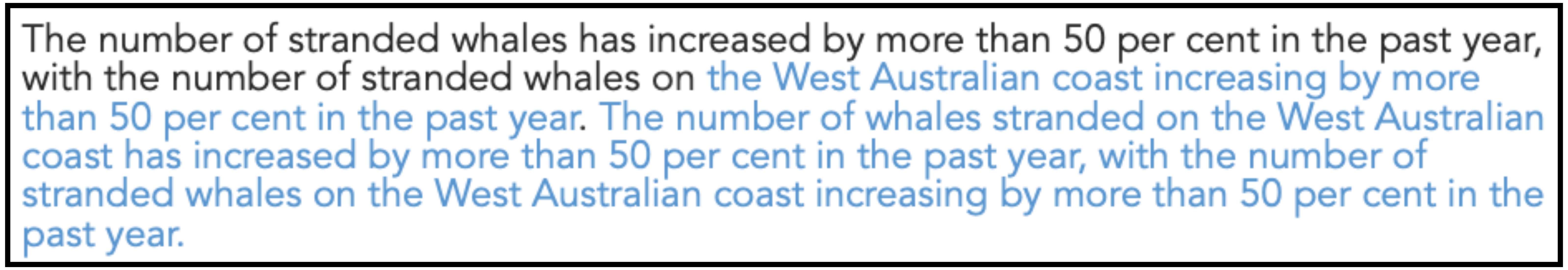
How can we generate more interesting and coherent text?

Issues with Greedy Decoding

[Holtzman et al., 2020]



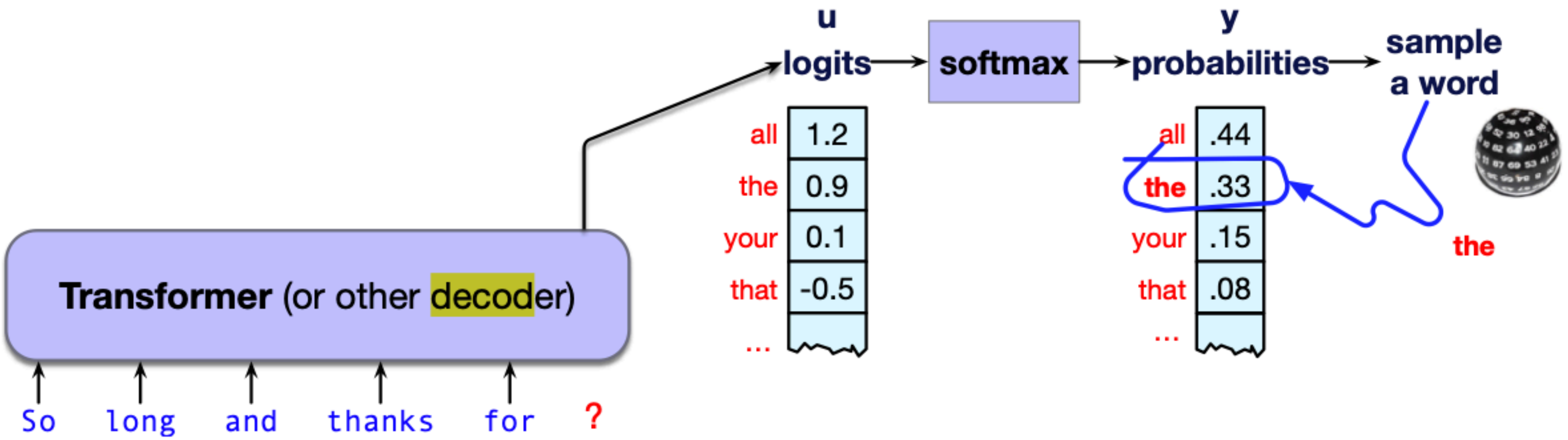
It is very easy to get **degenerate outputs** when using greedy decoding. The most likely tokens *right now* are not necessarily those that correspond to fluent language *overall*.



The number of stranded whales has increased by more than 50 per cent in the past year, with the number of stranded whales on the West Australian coast increasing by more than 50 per cent in the past year. The number of whales stranded on the West Australian coast has increased by more than 50 per cent in the past year, with the number of stranded whales on the West Australian coast increasing by more than 50 per cent in the past year.

(Pure) Sampling

The most common methods involve **sampling** from the LM's output distribution.

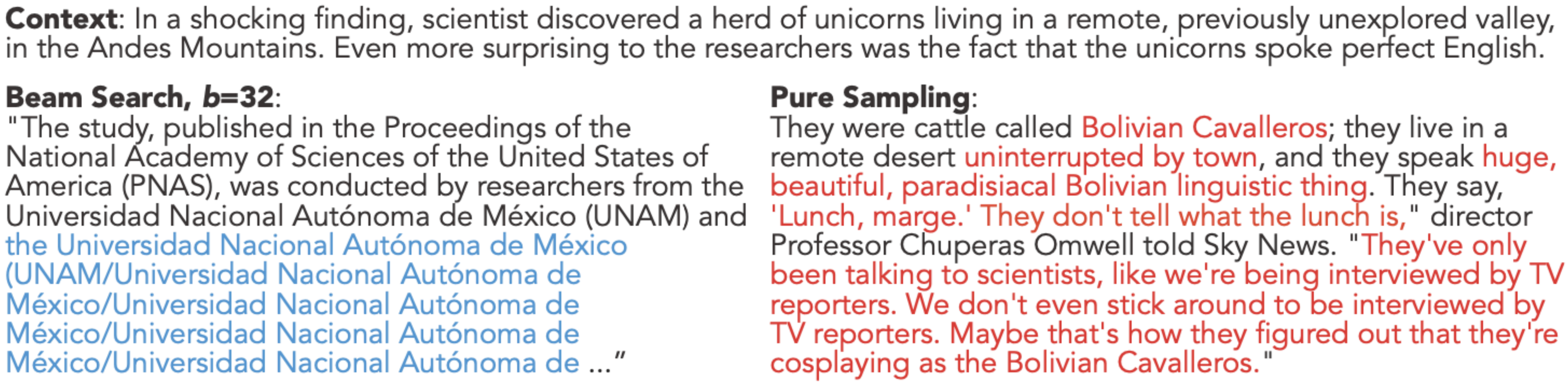


Just pick a token from the vocab at random—but according to the probabilities output by the LM.

This will generate mostly sensible outputs, but there are *a lot* of weird tokens in the tail of the probability distribution.

Issues with Sampling

[Holtzman et al., 2020]



Context: In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Beam Search, *b*=32:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

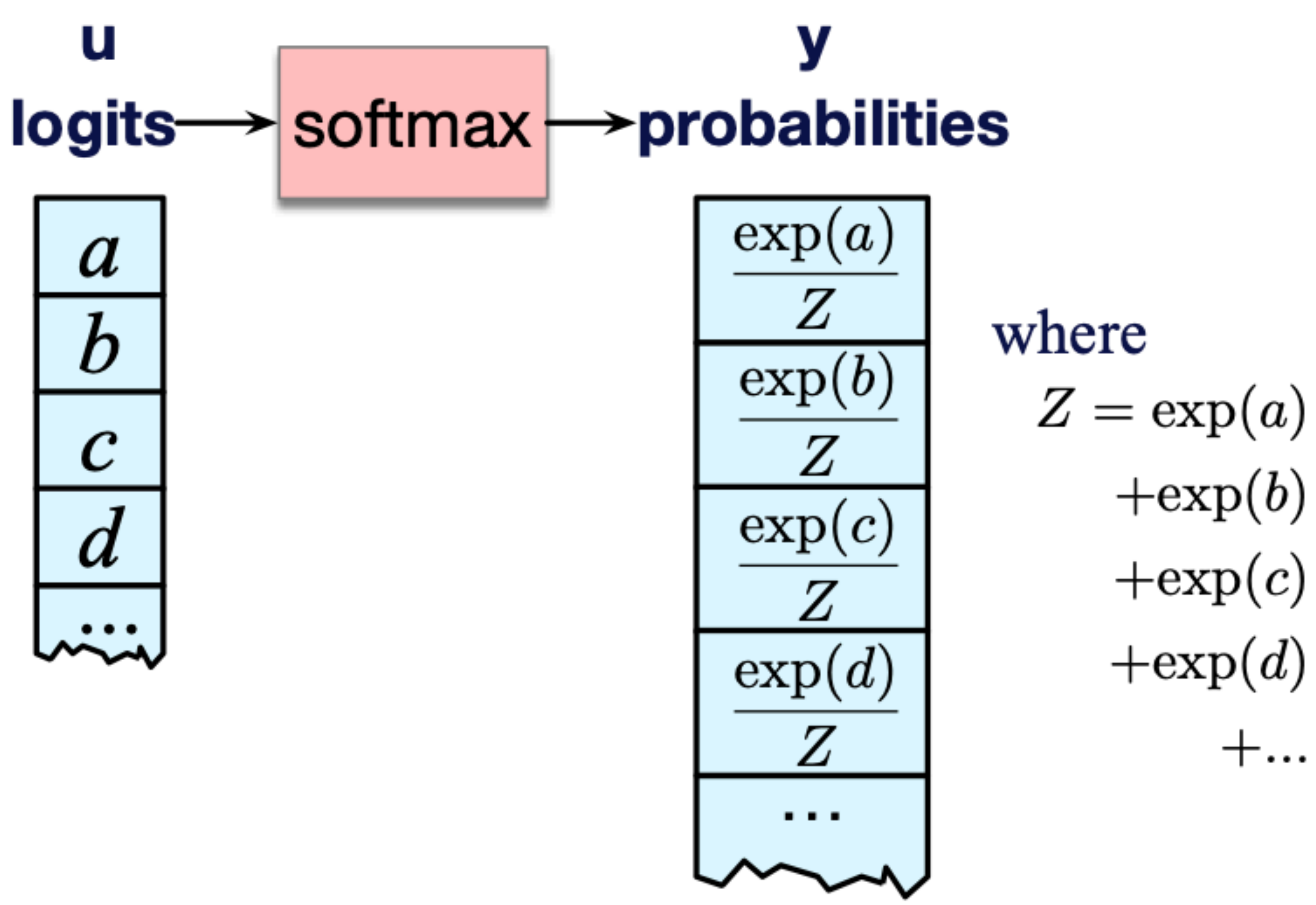
Even in a well-trained model, greedy decoding (and its improved cousin beam search) lead to degenerate outputs. Pure sampling leads to ungrammatical gibberish.

Temperature Sampling

Idea: reshape the probability distribution to increase the probability of high-probability tokens, and decrease the probability of low-probability tokens.

We use **temperature** τ, a hyperparameter, to do this.

$$\mathbf{y} = \text{softmax}(\mathbf{u}/\tau)$$



u

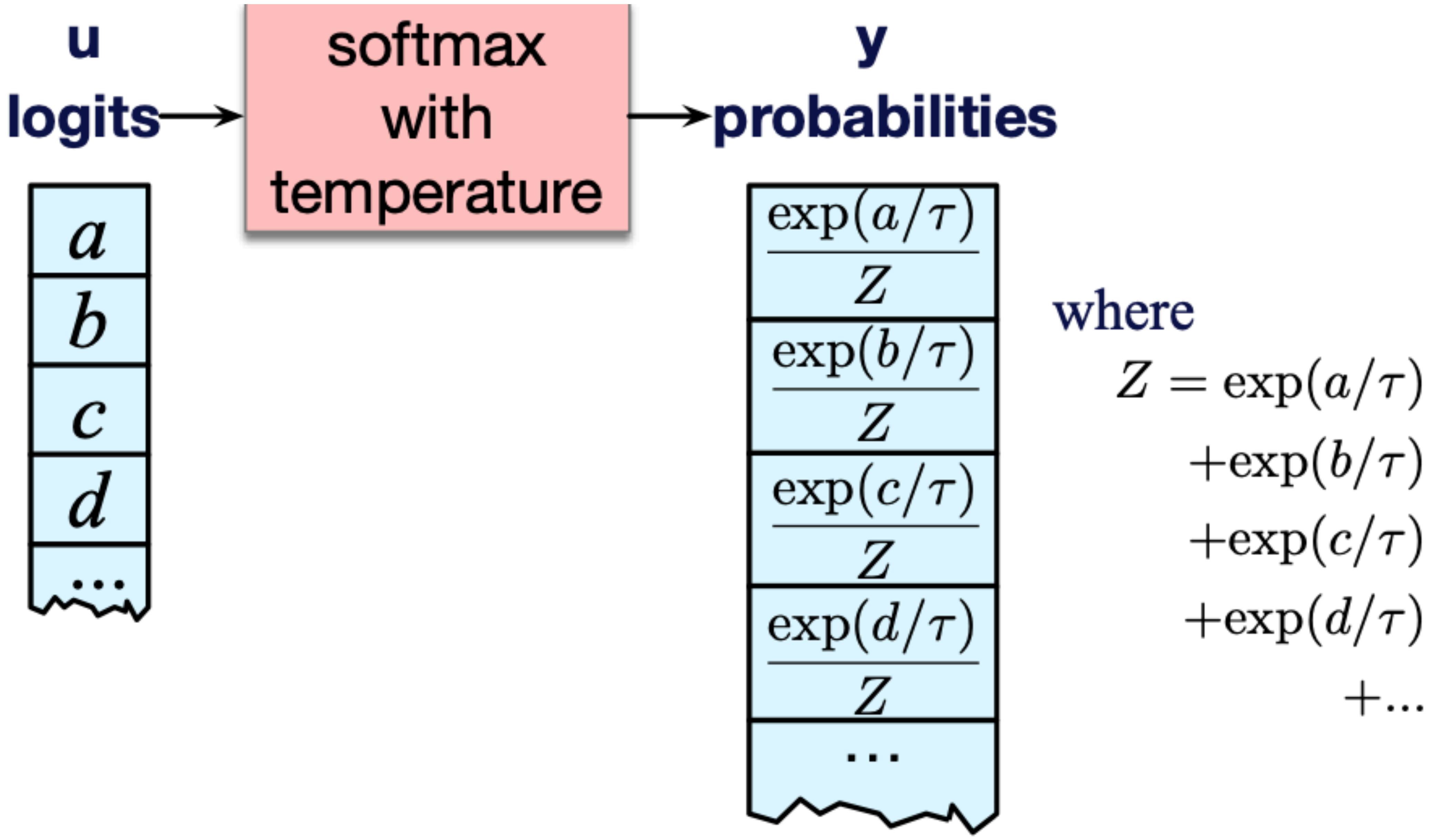
logits → softmax → **y** **probabilities**

a, b, c, d, …

$\frac{\exp(a)}{Z}$, $\frac{\exp(b)}{Z}$, $\frac{\exp(c)}{Z}$, $\frac{\exp(d)}{Z}$, …

where

$$Z = \exp(a) + \exp(b) + \exp(c) + \exp(d) + \ldots$$



u

logits → softmax with temperature → **y** **probabilities**

a, b, c, d, …

$\frac{\exp(a/\tau)}{Z}$, $\frac{\exp(b/\tau)}{Z}$, $\frac{\exp(c/\tau)}{Z}$, $\frac{\exp(d/\tau)}{Z}$, …

where

$$Z = \exp(a/\tau) + \exp(b/\tau) + \exp(c/\tau) + \exp(d/\tau) + \ldots$$

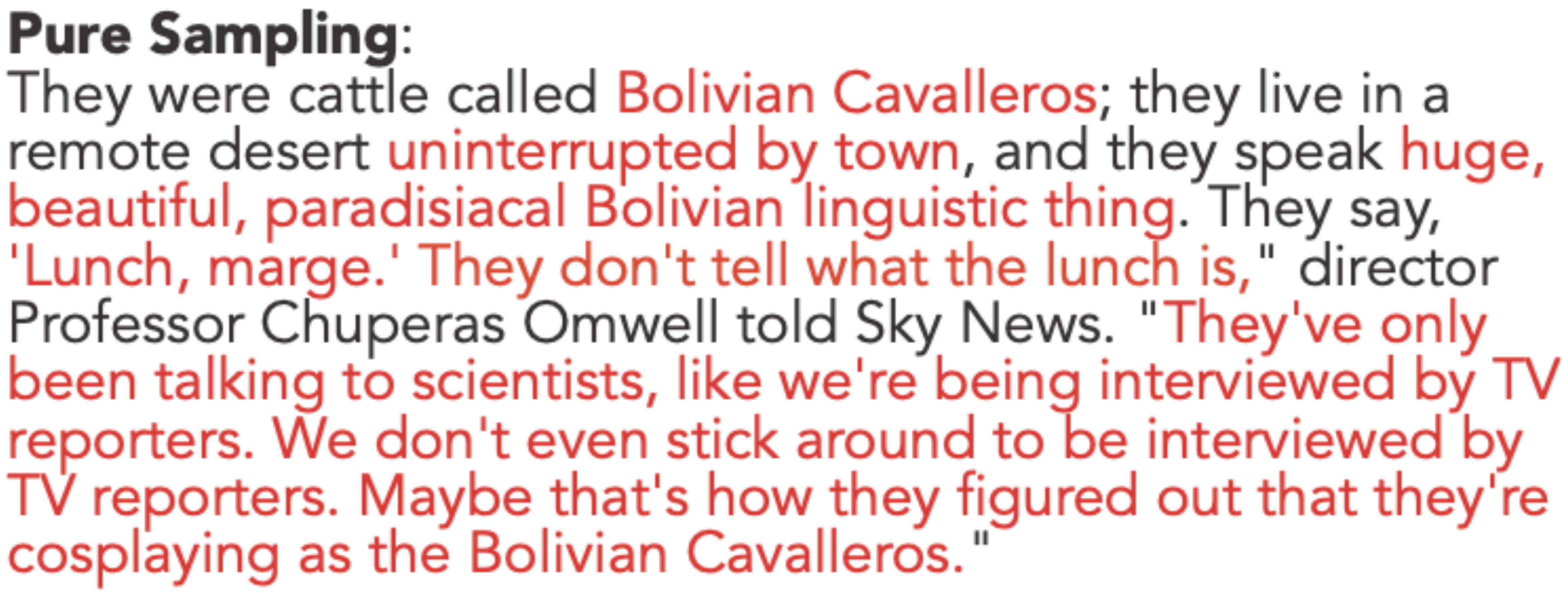
Temperature Sampling

- Low-temperature sampling ($\tau < 1$) makes more probable tokens even more probable, decreasing the “randomness” of the outputs.
    - $\tau = 0$ is the same as greedy decoding!
- High-temperature sampling ($\tau > 1$) increases the probability assigned to lower-probability tokens, increasing the “randomness” of the outputs.

	logits	τ=0.1	τ=0.5	τ=1	τ=10	τ=100
all	1.2	.95	.59	.44	.27	.25
the	0.9	.05	.32	.33	.26	.25
your	0.1	0	.07	.15	.24	.25
that	-0.5	0	.02	.08	.23	.25

Drawbacks of Pure Sampling

- Sampling is *too* random



Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisiacal Bolivian linguistic thing. They say, 'Lunch, marge.' They don't tell what the lunch is," director Professor Chuperas Omwell told Sky News. "They've only been talking to scientists, like we're being interviewed by TV reporters. We don't even stick around to be interviewed by TV reporters. Maybe that's how they figured out that they're cosplaying as the Bolivian Cavalleros."

$p(y \mid \ldots$and in the evenings, they enjoyed playing$)$

0.05	cards
0.01	basketball
0.01	Civilization
0.005	pretend
0.00005	JsonObj

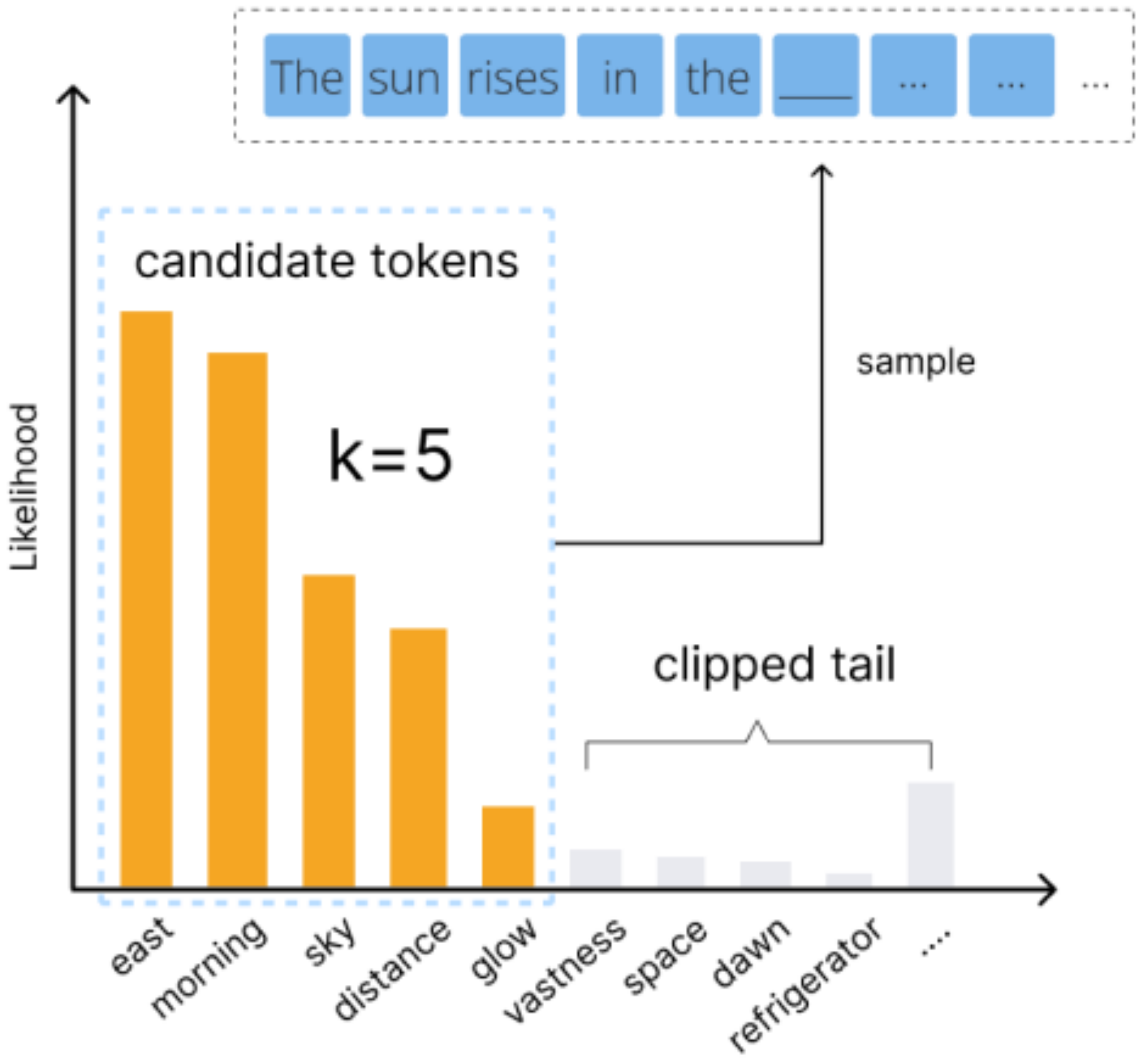
Top of distribution looks fine.

≈90% chance of getting something good.

Long tail of bad completions with ≈10% of the probability.

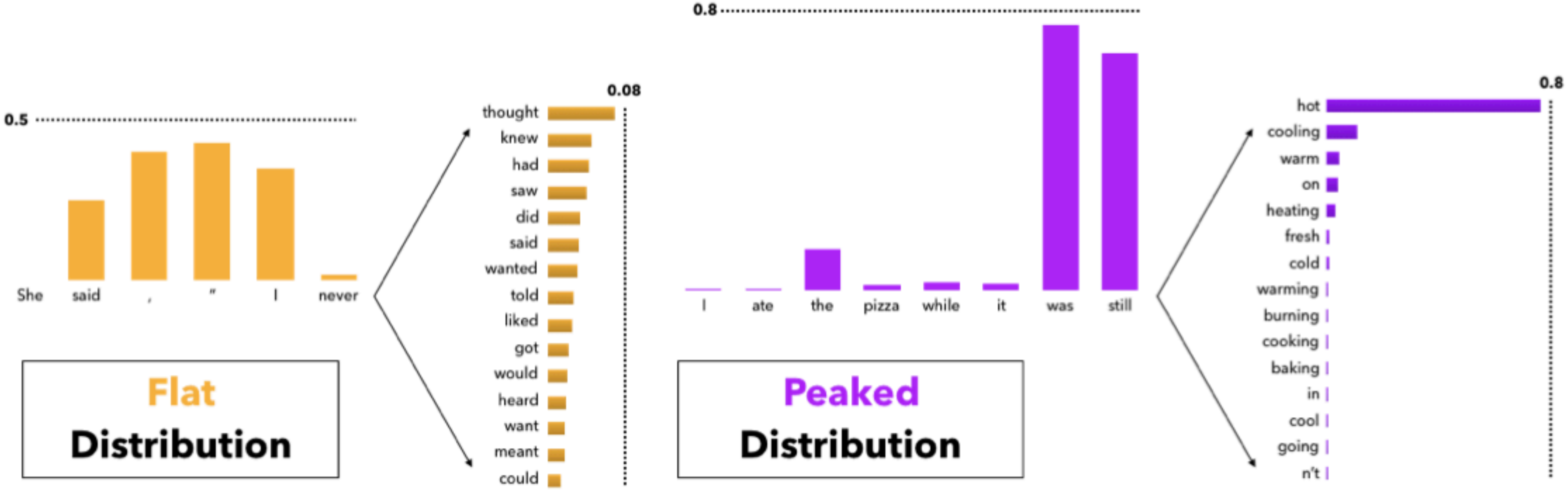
Top-k Sampling

Top-k sampling: We can truncate the distribution by sampling from *only* the top-k most probable tokens.



Nucleus Sampling

Nucleus (top-p) sampling: Or, we can sample from only the top tokens whose cumulative probability exceeds some threshold *p*.



(It's generally agreed that nucleus sampling is best.)

Nucleus Sampling

$p(y \mid \ldots$and in the evenings, they enjoyed playing$)$

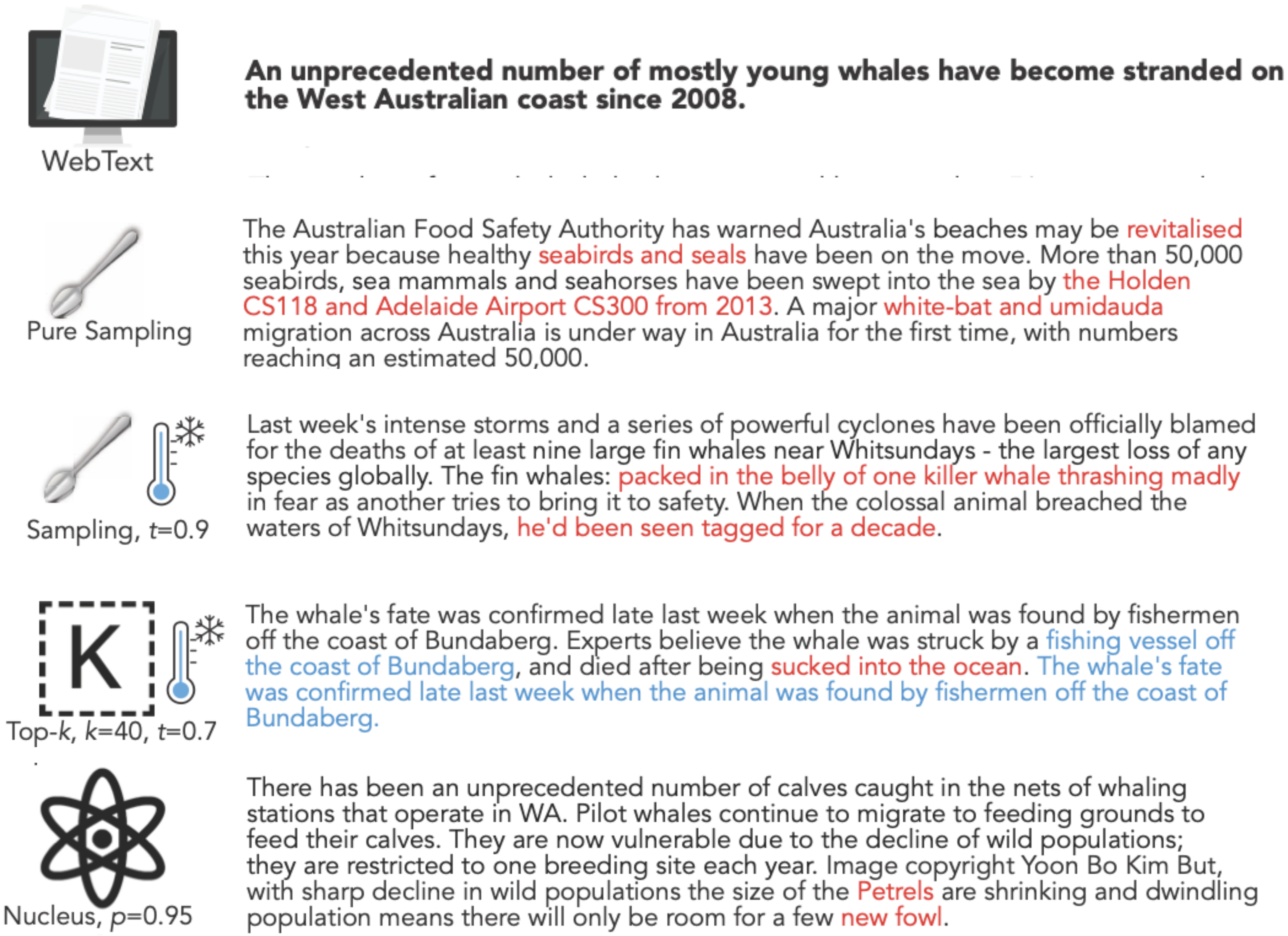
0.05	cards
0.01	basketball
0.01	Civilization
0.005	pretend

Sample from top of distribution only.

Cut off distribution after *p*% of probability mass.

- Keep the most probable options accounting for *p*% of the probability mass (the “nucleus”), and sample only among these.
- To implement: sort tokens by probability, truncate list as soon as you exceed *p,* then renormalize and sample from that new distribution

Examples



WebText

An unprecedented number of mostly young whales have become stranded on the West Australian coast since 2008.

Pure Sampling

The Australian Food Safety Authority has warned Australia's beaches may be revitalised this year because healthy seabirds and seals have been on the move. More than 50,000 seabirds, sea mammals and seahorses have been swept into the sea by the Holden CS118 and Adelaide Airport CS300 from 2013. A major white-bat and umidauda migration across Australia is under way in Australia for the first time, with numbers reaching an estimated 50,000.

Sampling, *t*=0.9

Last week's intense storms and a series of powerful cyclones have been officially blamed for the deaths of at least nine large fin whales near Whitsundays - the largest loss of any species globally. The fin whales: packed in the belly of one killer whale thrashing madly in fear as another tries to bring it to safety. When the colossal animal breached the waters of Whitsundays, he'd been seen tagged for a decade.

Top-*k*, *k*=40, *t*=0.7

The whale's fate was confirmed late last week when the animal was found by fishermen off the coast of Bundaberg. Experts believe the whale was struck by a fishing vessel off the coast of Bundaberg, and died after being sucked into the ocean. The whale's fate was confirmed late last week when the animal was found by fishermen off the coast of Bundaberg.

Nucleus, *p*=0.95

There has been an unprecedented number of calves caught in the nets of whaling stations that operate in WA. Pilot whales continue to migrate to feeding grounds to feed their calves. They are now vulnerable due to the decline of wild populations; they are restricted to one breeding site each year. Image copyright Yoon Bo Kim But, with sharp decline in wild populations the size of the Petrels are shrinking and dwindling population means there will only be room for a few new fowl.

Distributional Semantics, redux

Recall our old friend, distributional semantics:

> “You shall know a word by the company it keeps.” —**Firth, 1957**

You can never fully characterize the distribution of a word: more data will always give you more accurate estimates.

- Also, word distributions are always changing because language is always changing!

Given this, it seems like more data will always be better. *This is empirically what we’ve observed over the last decade*.

The Unreasonable Effectiveness of Pretraining

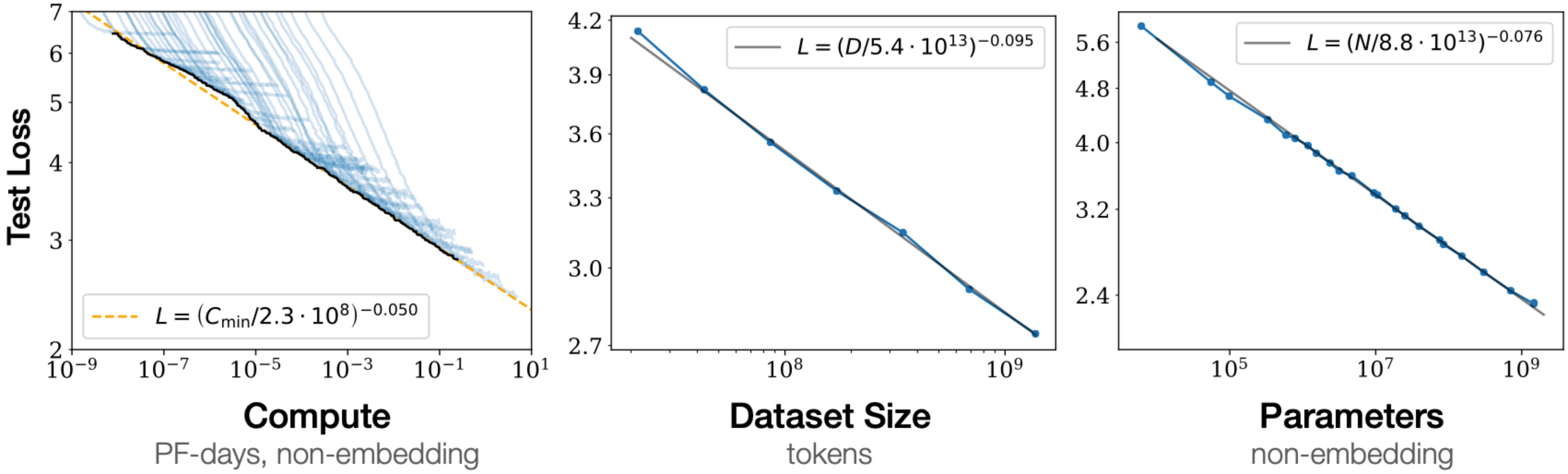
Reading Comprehension
F1
90
80
70
60
50
40
30
Human
DrQA+PGNet
DrQA
PGNet
Seq2seq
117M
345M
762M
1542M
# of parameters in LM
Translation
BLEU
25
20
15
10
5
0
Unsupervised Statistical MT
Denoising + Backtranslate
Embed Nearest Neighbor
Denoising
117M
345M
762M
1542M
# of parameters in LM
Summarization
Average of ROUGE 1,2,L
32
30
28
26
24
22
20
18
16
Lead-3
PGNet
Seq2seq + Attn
Random-3
117M
345M
762M
1542M
# of parameters in LM
Question Answering
Accuracy
10
8
6
4
2
0
↑↑Open Domain QA Systems↑↑
most freq Q-type answer
117M
345M
762M
1542M
# of parameters in LM

As we train larger and larger models, performance on *any* NLP task seems to increase.

It turns out that there are *predictable* improvements as we increase dataset size and model size!

Scaling Laws

[Kaplan et al., 2020]

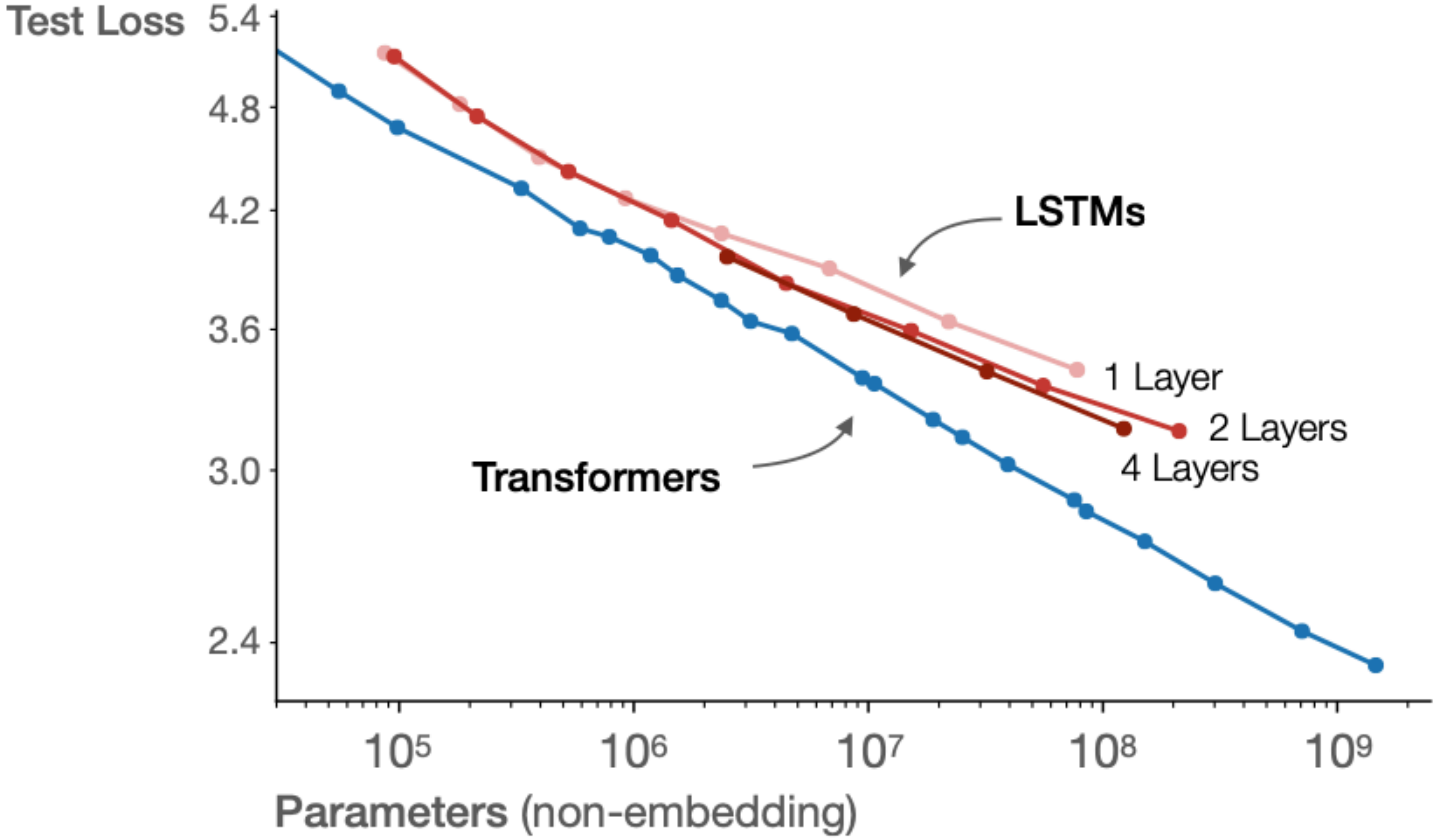


Scaling laws yield simple-yet-predictive rules for model performance w.r.t. increasing scale (of data and models).

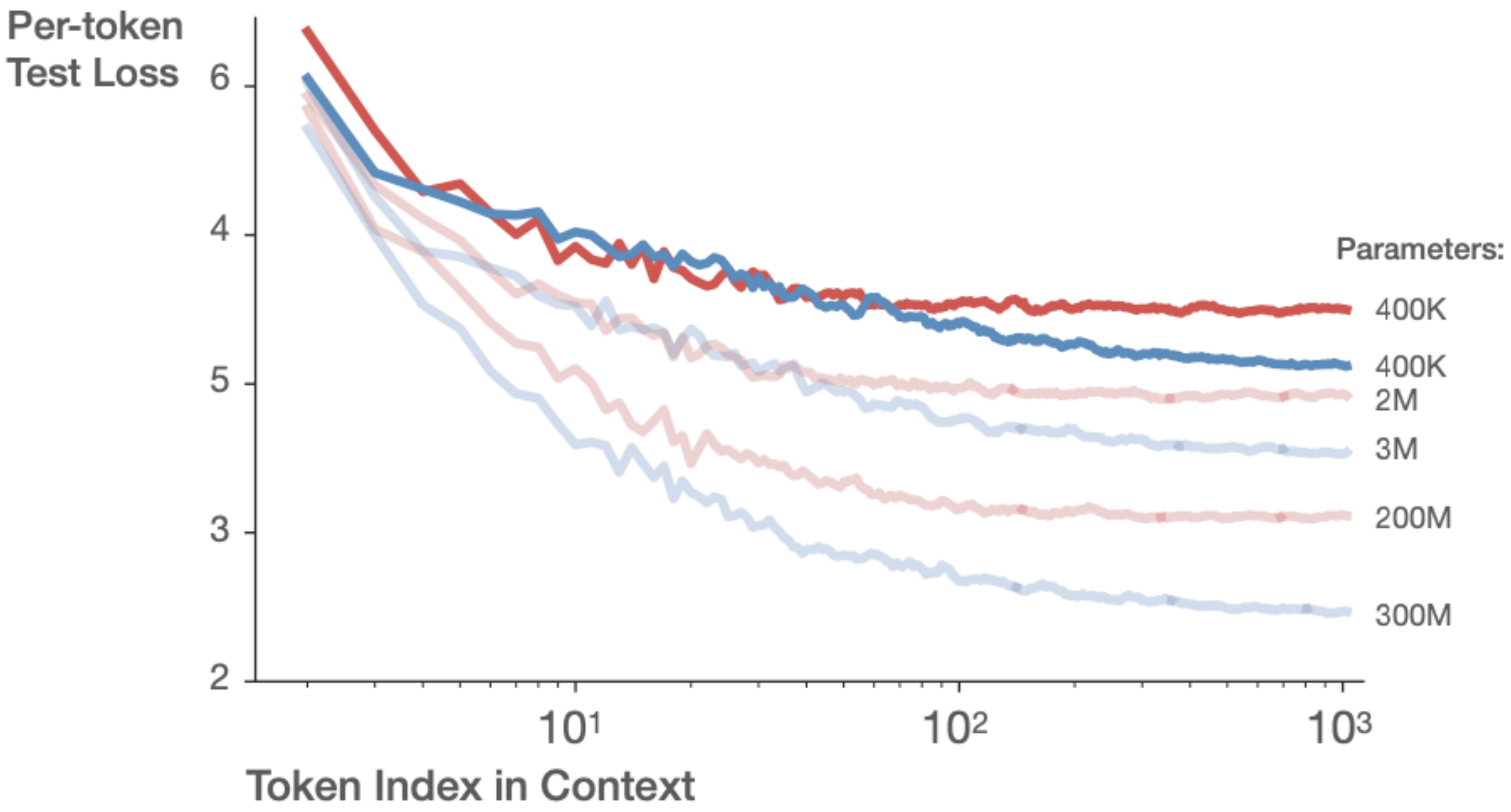
Big implications: innovate at small scale, and extrapolate to larger versions of that model.

Scaling Laws

Transformers asymptotically outperform LSTMs due to improved use of long contexts



LSTM plateaus after <100 tokens
Transformer improves through the whole context

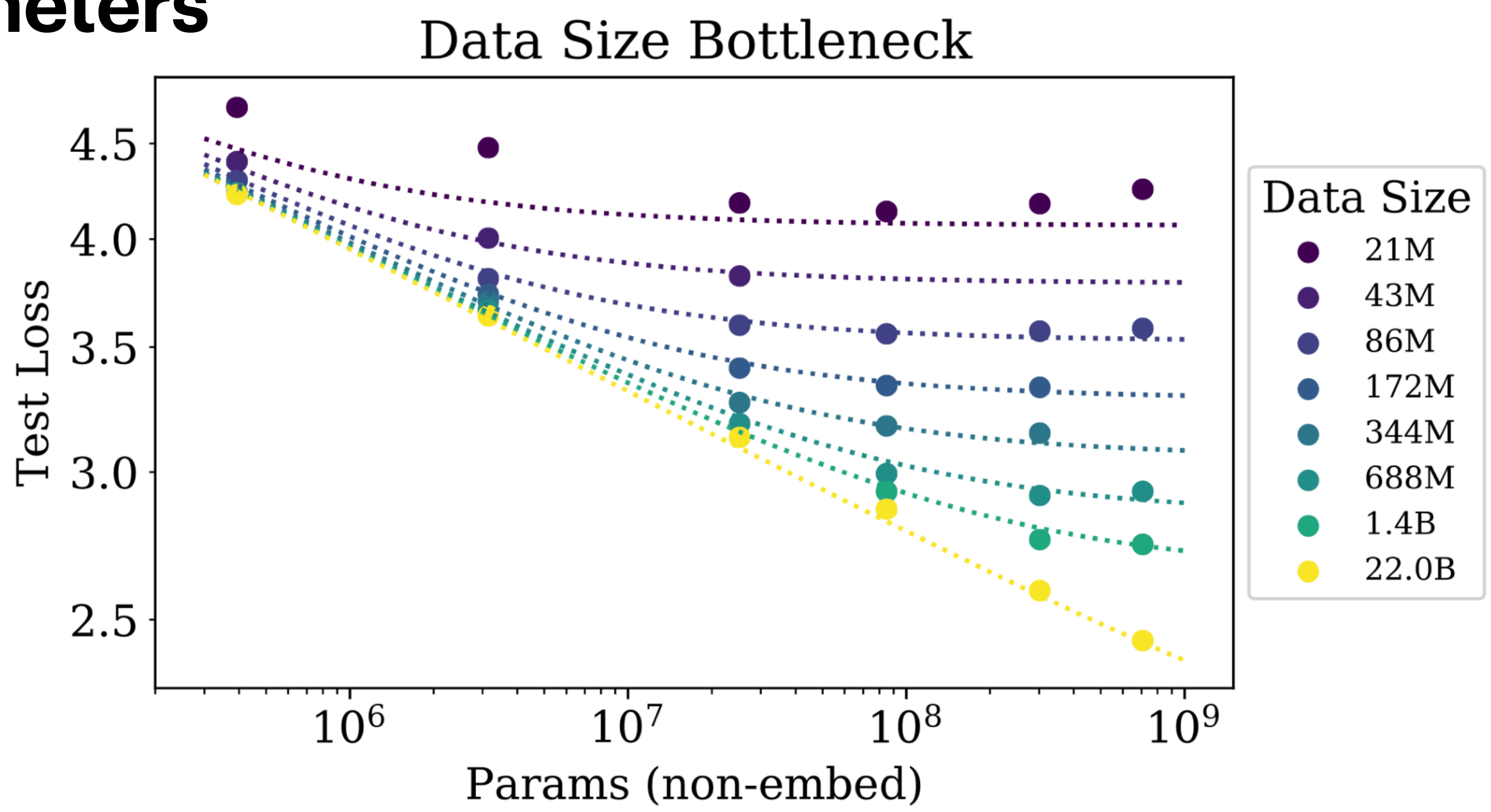


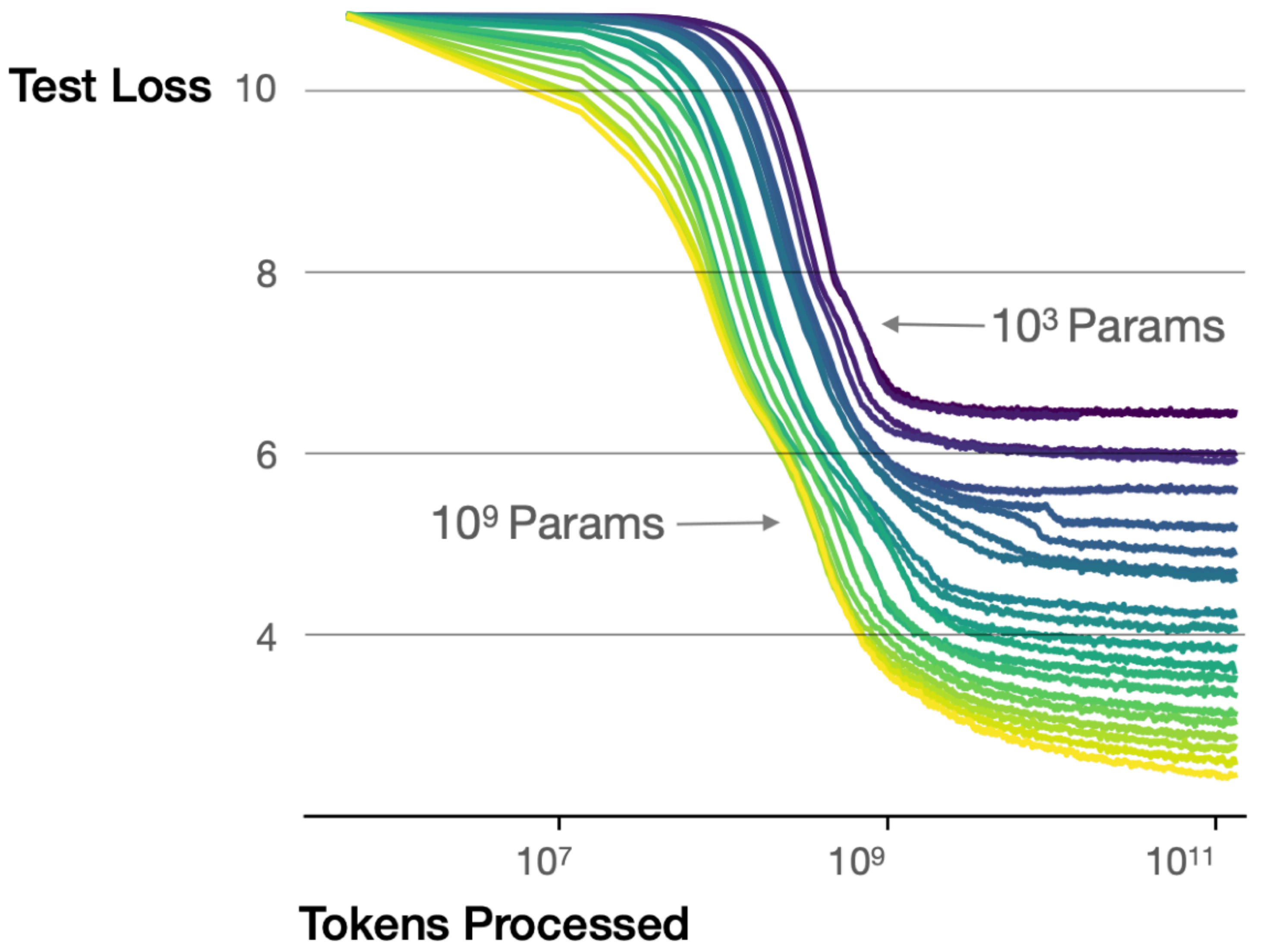
Scaling Laws

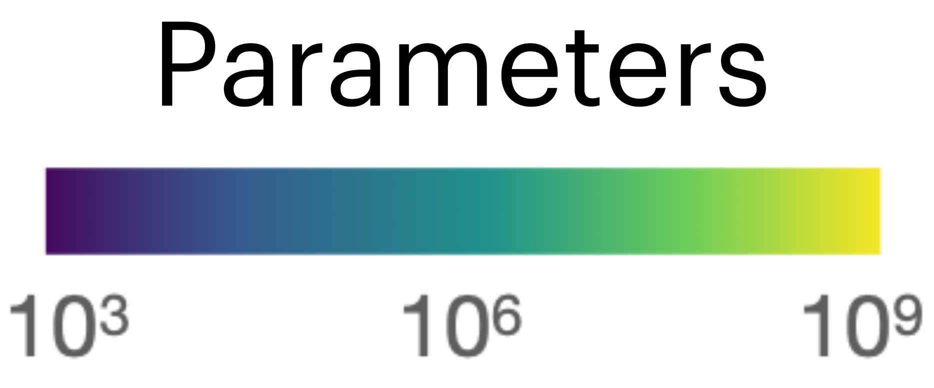
Data vs. Parameters

- As number of parameters increases, the amount of data needed to achieve close to a model's optimal loss also increases.

- However, larger models require less data to get a similar loss to smaller models!







Scaling Laws

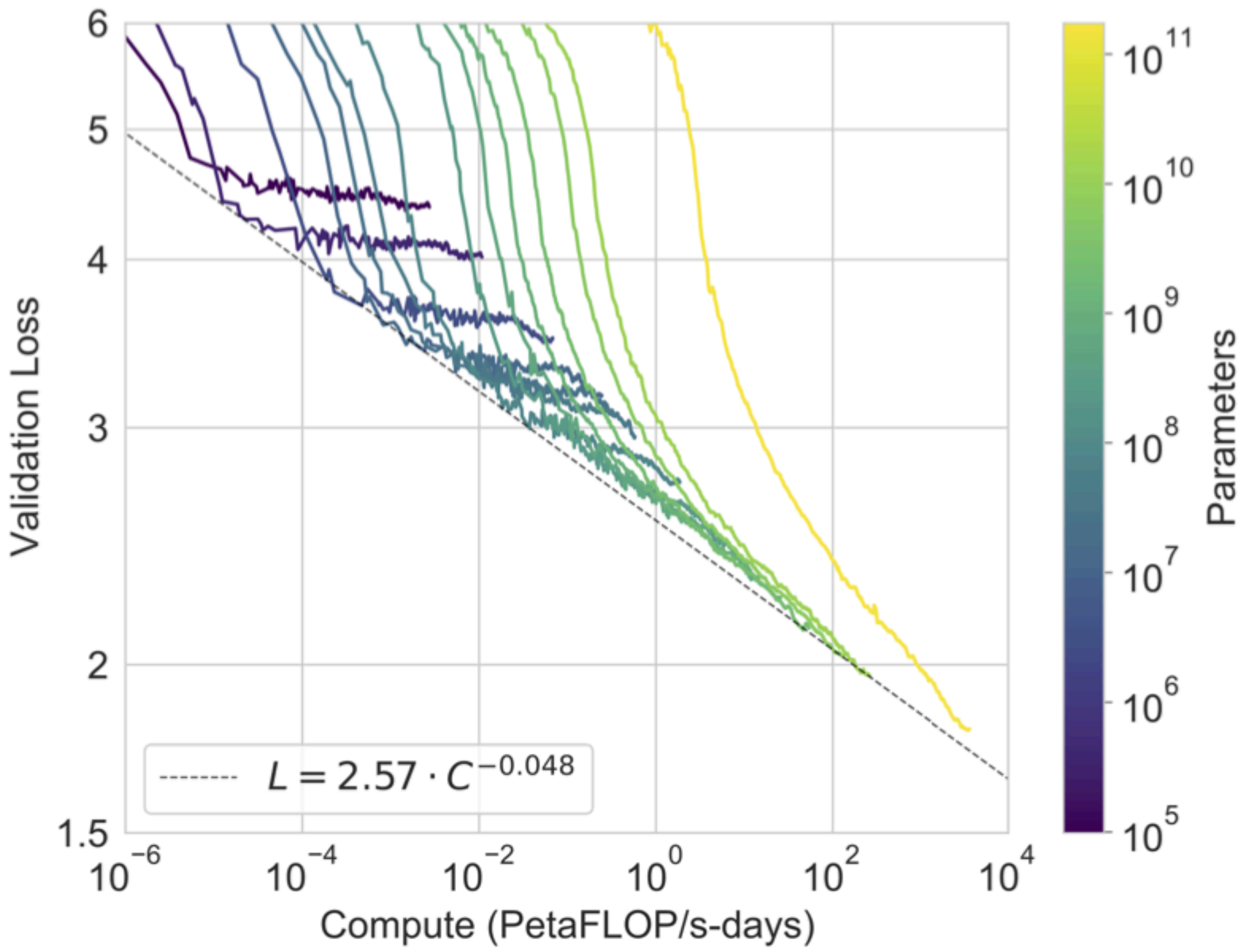
Compute vs. Loss

Larger models generally require more compute to get to an optimal loss.

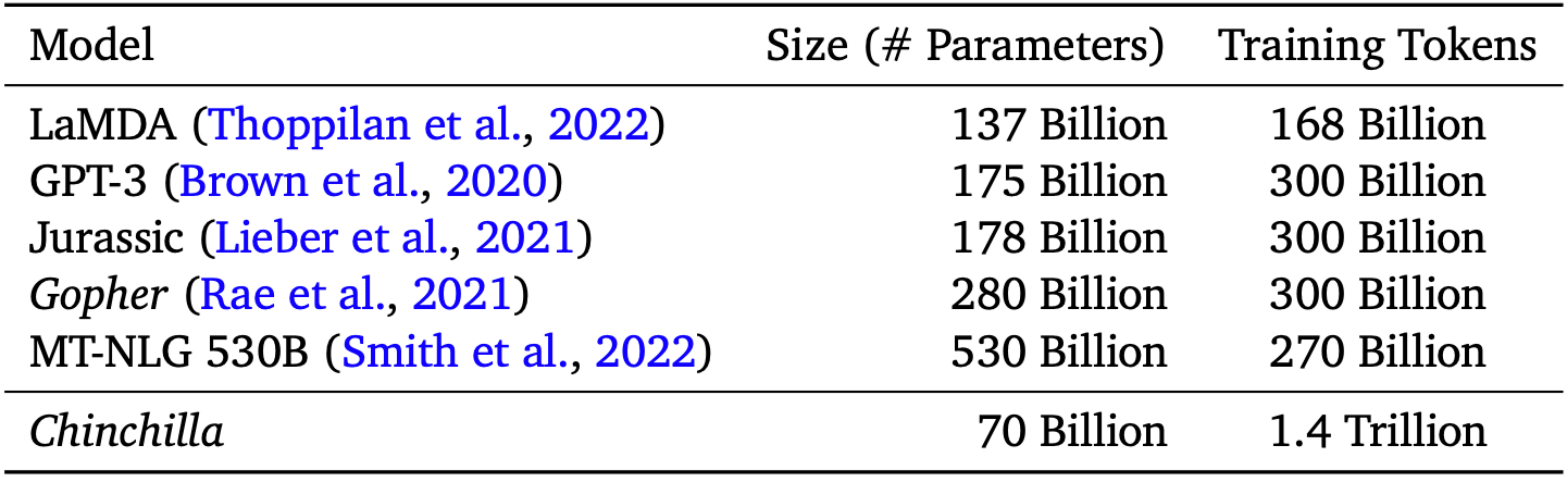
The best loss one can achieve generally improves as a sublinear function of compute:

$$L = 2.57 \cdot C^{-0.048}$$

I.e., for a ≈10% improvement in loss, you need ≈10x compute.



Scaling Efficiency



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
Chinchilla	70 Billion	1.4 Trillion

Bigger is not guaranteed to be better!

Chinchilla, a 70B-parameter model, outperforms models more than twice its size.

GPT-3 and Beyond

- Depending on who you ask, the era of “large” language models (LLMs) started with either:
  - 2018: GPT - first Transformer-based LM trained on a decently large corpus
  - 2020: GPT-3 - first general-purpose LM that achieved great performance on many tasks

- Once you reach a certain point, a model gains the hugely useful ability to do things like **in-context learning**.
  - That is, learning from just text without any gradient updates. (More on this later.)
- The first model that was consistely able to do in-context learning well was GPT-3.
  - 175 billion parameters!

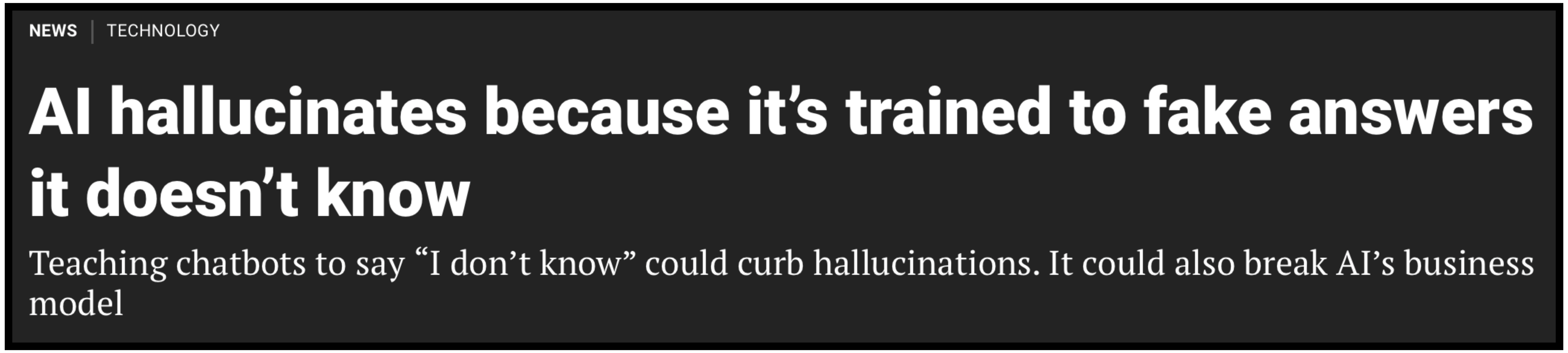
If you can learn a task in-context, do you even need fine-tuning?

Perplexity isn’t everything.

- **Size:** bigger models take a lot of time and GPUs to train, and a lot of memory to store.

- **Energy usage:** larger models/more tokens require more energy, and thus usually lead to more CO2 emissions
    - Water usage (for cooling data center servers) has also been controversial recently

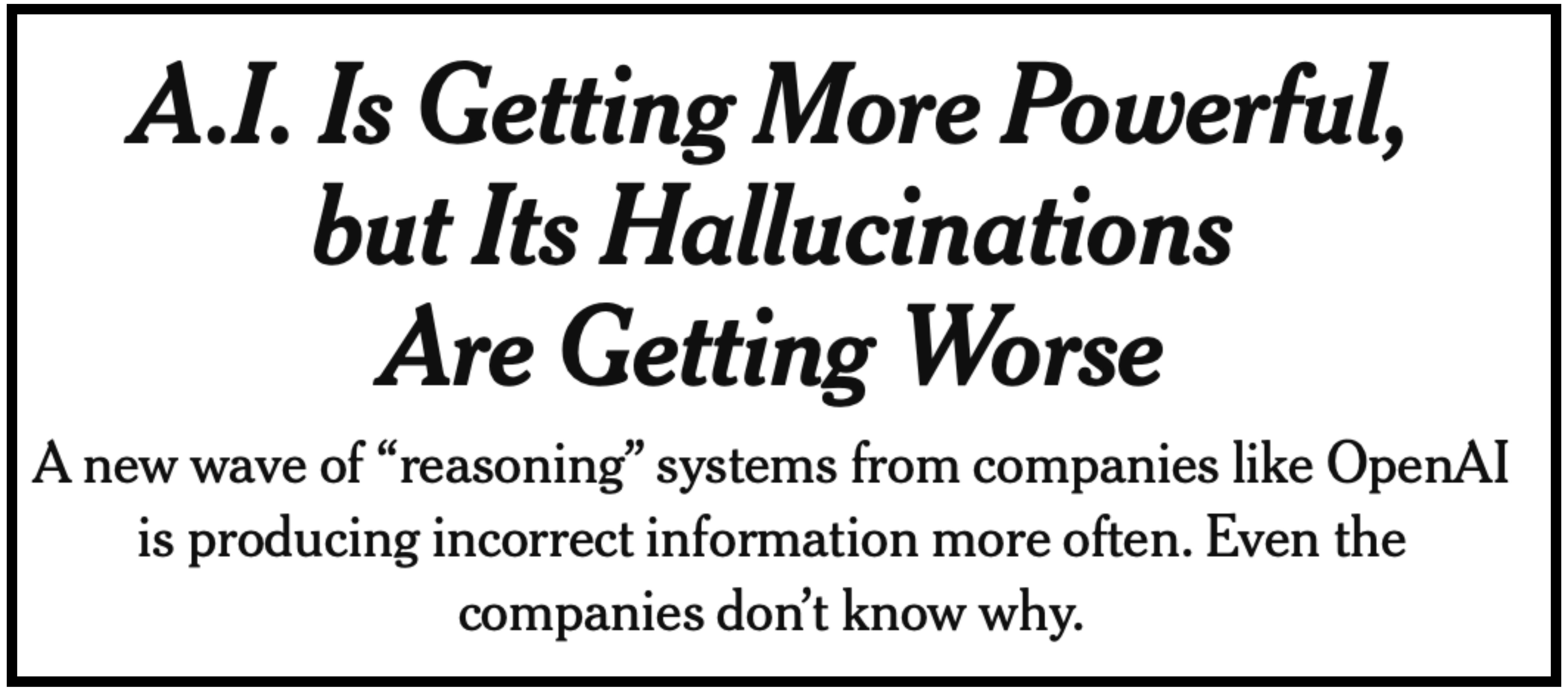
- **Fairness:** even when task performance is good, language models can often pick up on stereotypes, or have systematically lower performance when handling certain dialects or information from certain demographic groups



NEWS | TECHNOLOGY

AI hallucinates because it’s trained to fake answers it doesn’t know

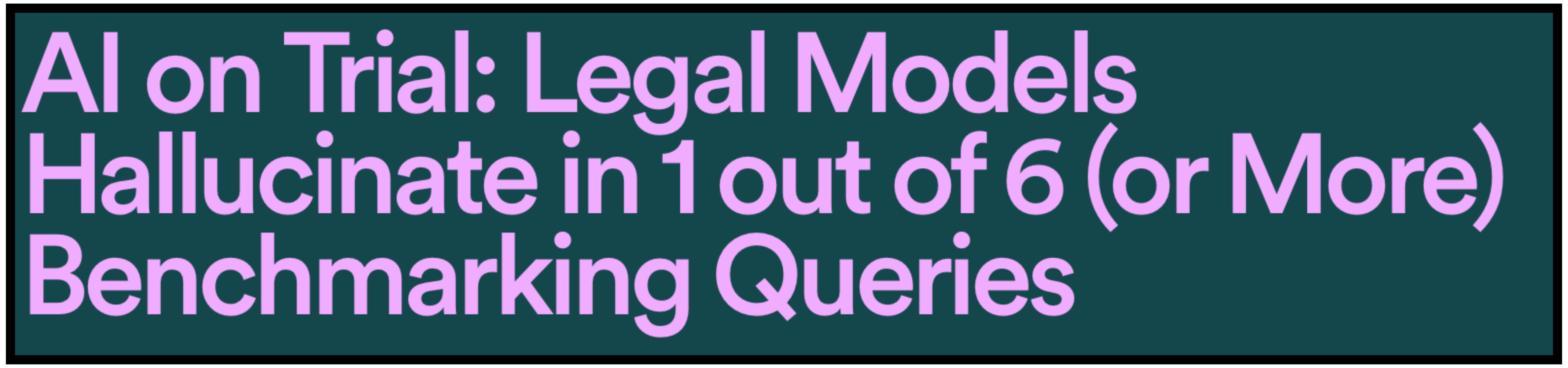
Teaching chatbots to say “I don’t know” could curb hallucinations. It could also break AI’s business model



A.I. Is Getting More Powerful, but Its Hallucinations Are Getting Worse

A new wave of “reasoning” systems from companies like OpenAI is producing incorrect information more often. Even the companies don’t know why.

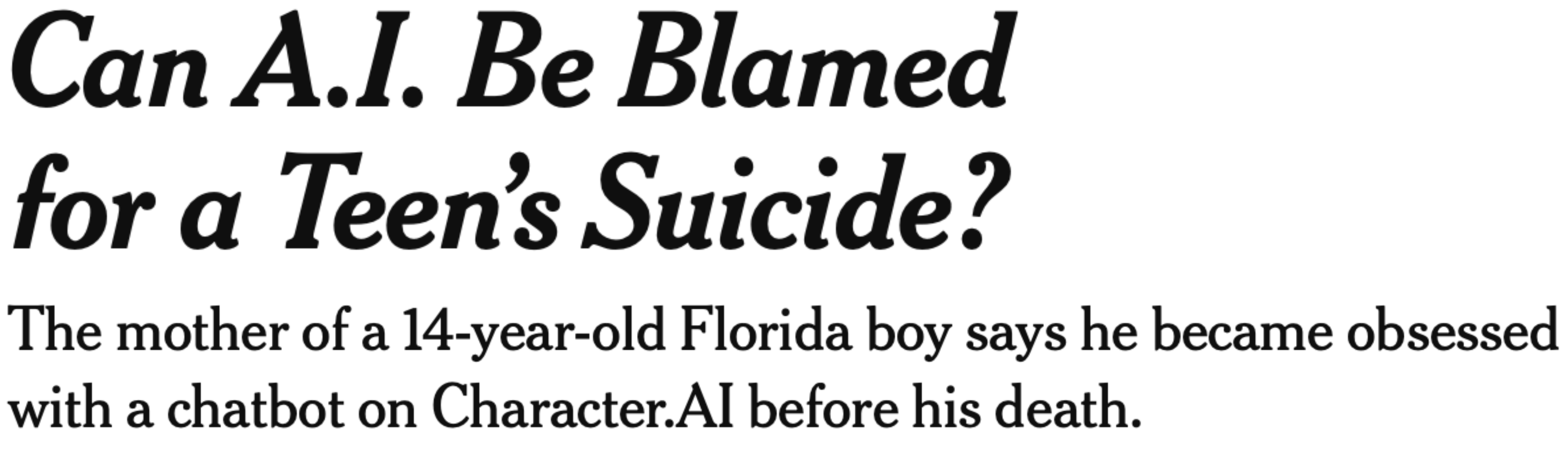
Hallucination: When an LLM generates factually incorrect or nonsensical information (often with a confident tone).



AI on Trial: Legal Models Hallucinate in 1 out of 6 (or More) Benchmarking Queries

Ethical Issues with LLMs

- **Unsafe suggestions:** LLMs can suggest that users do dangerous or illegal things
- **Syncophancy:** LLMs often blindly agree with everything a user says
- **Emotional dependence:** people tend to assign human characteristics to computers **[Reeves & Nass, 1996]** and as such can impact people's emotional and cognitive states
- **Fraud assistance**: LLMs make it easier for bad actors to do things like phishing, create propaganda and misinformation, amplify extreme views, etc.



Can A.I. Be Blamed for a Teen's Suicide?

The mother of a 14-year-old Florida boy says he became obsessed with a chatbot on Character.AI before his death.

AI could accelerate scientific fraud as well as progress

Efficiency

What do we mean when we say that a system learns efficiently?

- **Sample (data) efficiency**: It can learn with fewer examples than other systems.
- **Time efficiency**: It can learn in little (wall clock) time.
- **Compute efficiency**: It can learn with fewer floating-point operations (FLOPs).

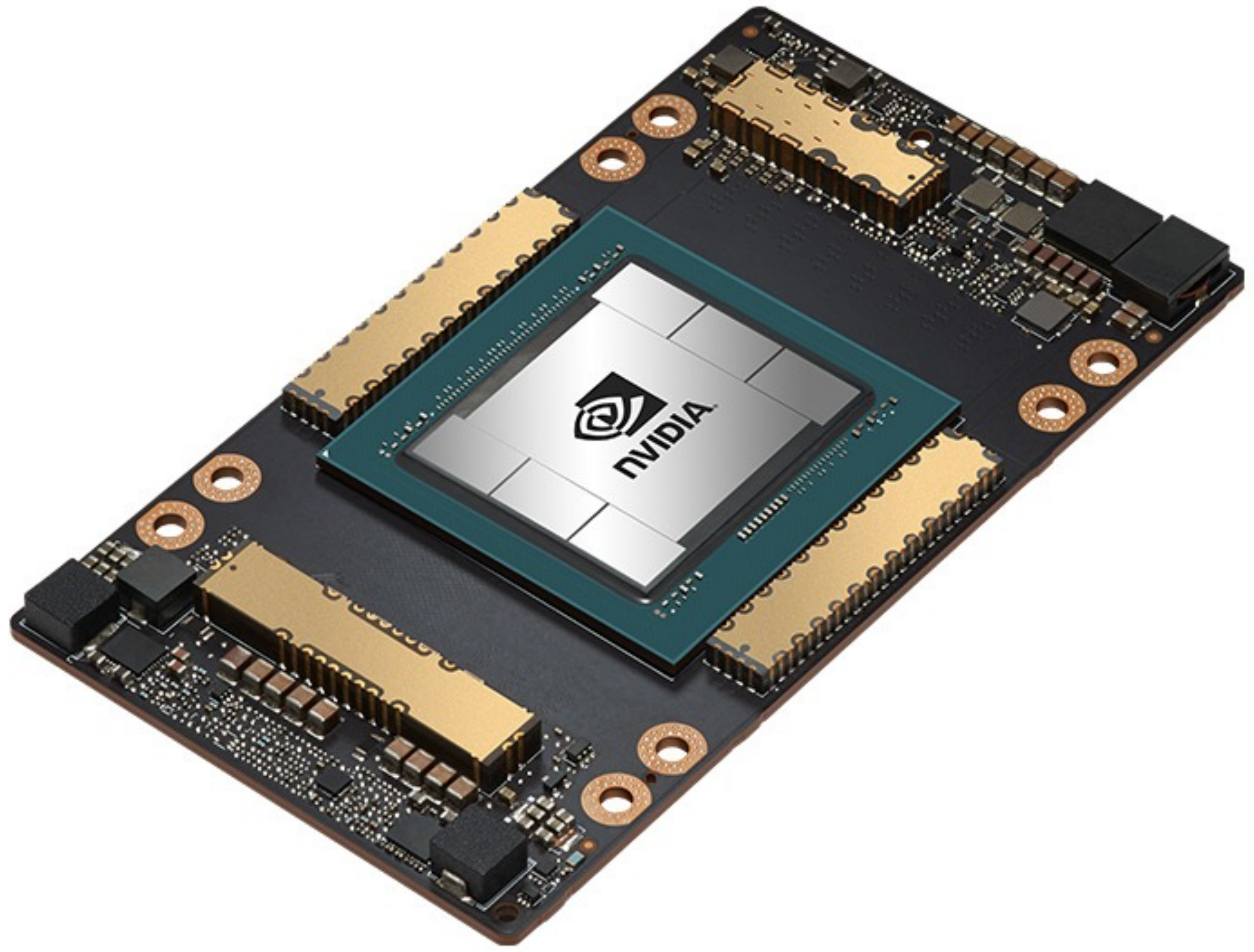
It's hard to standardize measurements of time; this depends on a lot on your hardware, what else is running in the background, etc.

We often measure compute in terms of how many examples the model must process, and how large the model is:

$$\text{Computaton} \propto \text{NumParameters} \times \text{NumExamples}$$

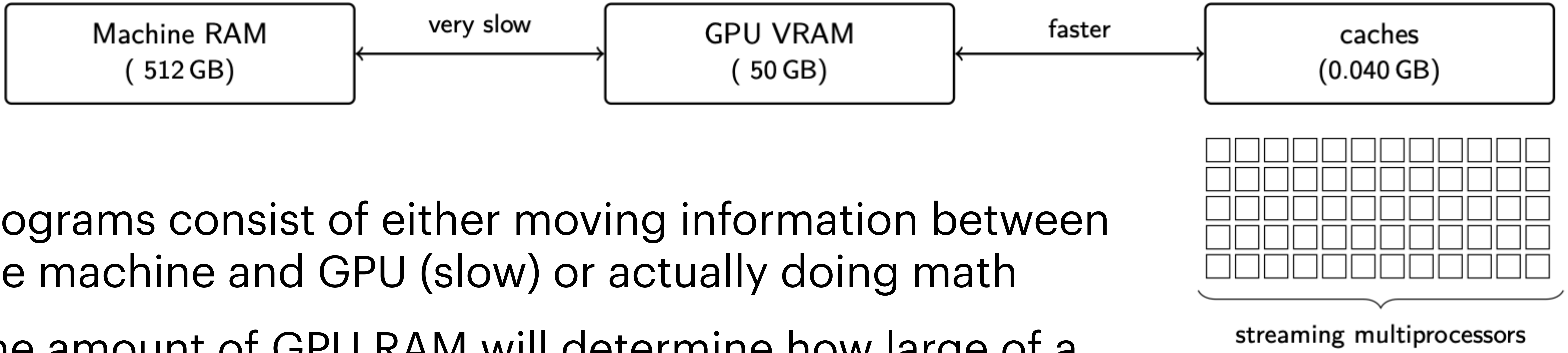
Parallelizability

- Time-wise, NumParameters x NumExamples isn’t the whole story: RNNs will train *far* more slowly than a Transformer given the same number examples.
- This is because Transformers are far more parallelizable: they can train on many tokens in parallel. An RNN must process sequences token-by-token.
- Parallelization only helps if we have access to hardware that enables parallel computation.
- Enter: GPUs and TPUs!



GPUs

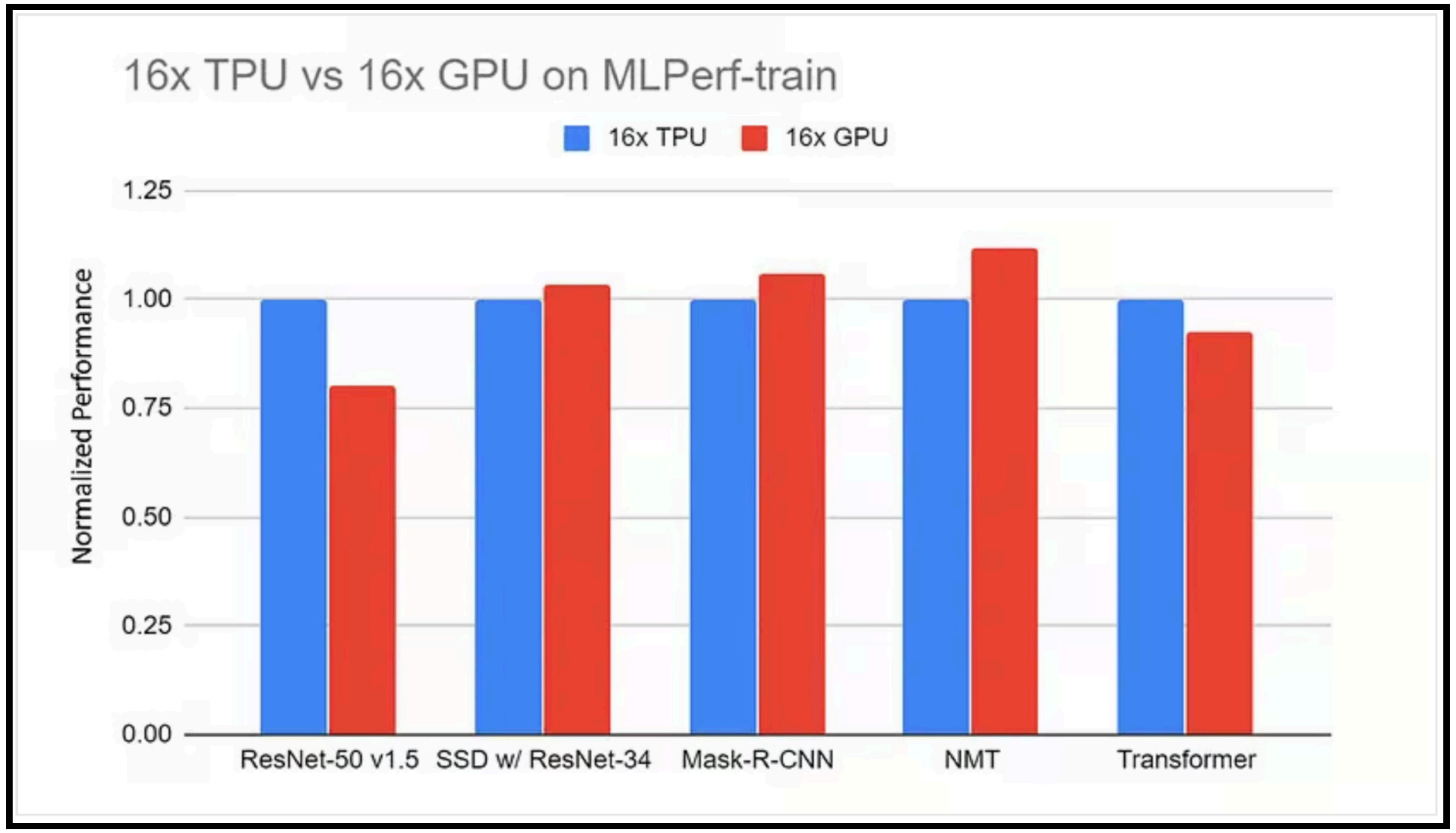
- GPUs are a huge reason why neural networks are so popular today.
- A GPU is a massively parallel processor with its own memory and cores.



- Programs consist of either moving information between the machine and GPU (slow) or actually doing math
- The amount of GPU RAM will determine how large of a model you can reasonably work with

TPUs

- GPUs were designed for graphics processing, but later retrofitted for AI.
- Most of the parallelizable parts of AI involve **tensor** operations.
- TPUs are specifically designed to make tensor operations fast.



[Khairy, 2020]

Some GPU Heuristics

- If you have a GPU with **40GB of VRAM**, you can probably do the following:
    - Generate (do “inference”) with a model of < 13B - 20B parameters
        - Up to 70B - 80B parameters if you reduce precision and work with short sequences
    - Fine-tune a model of < 3B - 7B parameters
    - Train a model of < 500M - 1B parameters
        - This will take forever! You’d probably want to reduce the number of parameters so you can increase the batch size.