

Large Language Models

Introduction to Large Language Models

Large Language Models

 **OpenAI GPT-5**

ANTHROPIC CLAUDE

 deepseek
R1

 Gemini

Microsoft[®] PHI-4

 **Meta**
LLAMA

Language models

Remember the simple **n-gram language model**

- Assigns probabilities to sequences of words
- Generate text by sampling possible next words
- Is trained on counts computed from lots of text

Large language models are similar and different:

- Assigns probabilities to sequences of words
- Generate text by sampling possible next words
- **Are trained by learning to guess the next word**

Fundamental intuition of large language models

Text contains enormous amounts of knowledge

Pretraining on lots of text with all that knowledge is what gives language models their ability to do so much

What does a model learn from pretraining?

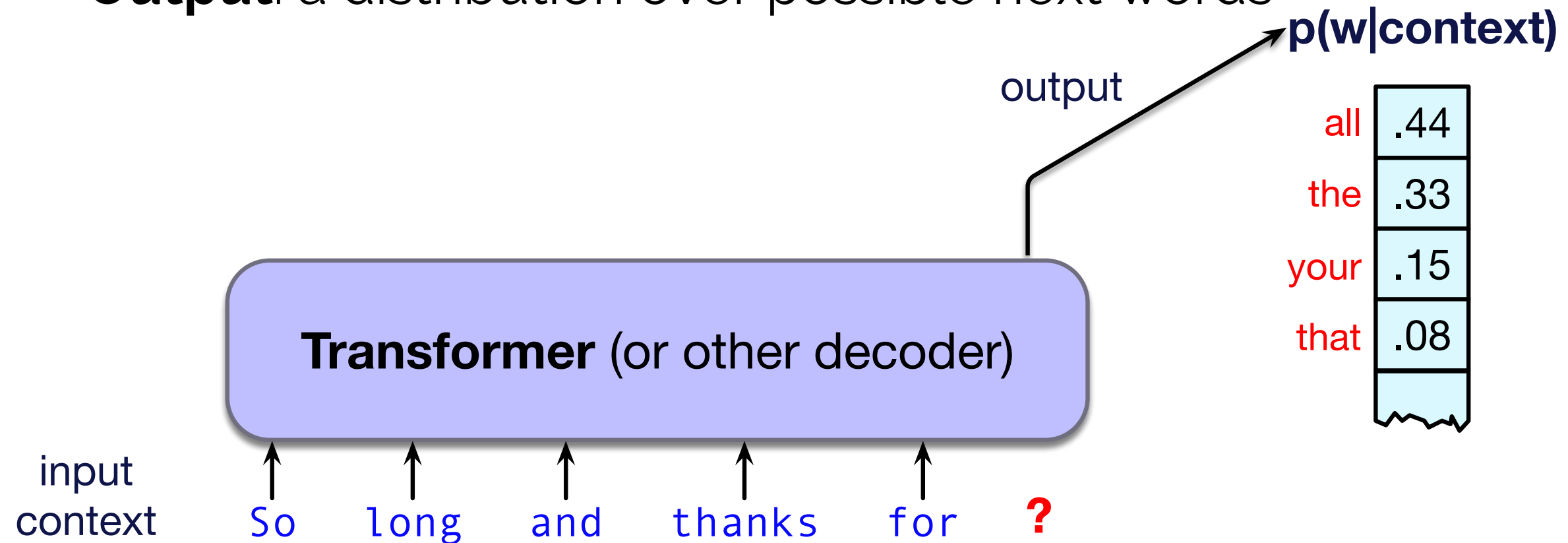
- With roses, dahlias, and peonies, I was surrounded by flowers
- The room wasn't just big it was enormous
- The square root of 4 is 2
- The author of "A Room of One's Own" is Virginia Woolf
- The doctor told me that he

What is a large language model?

A neural network with:

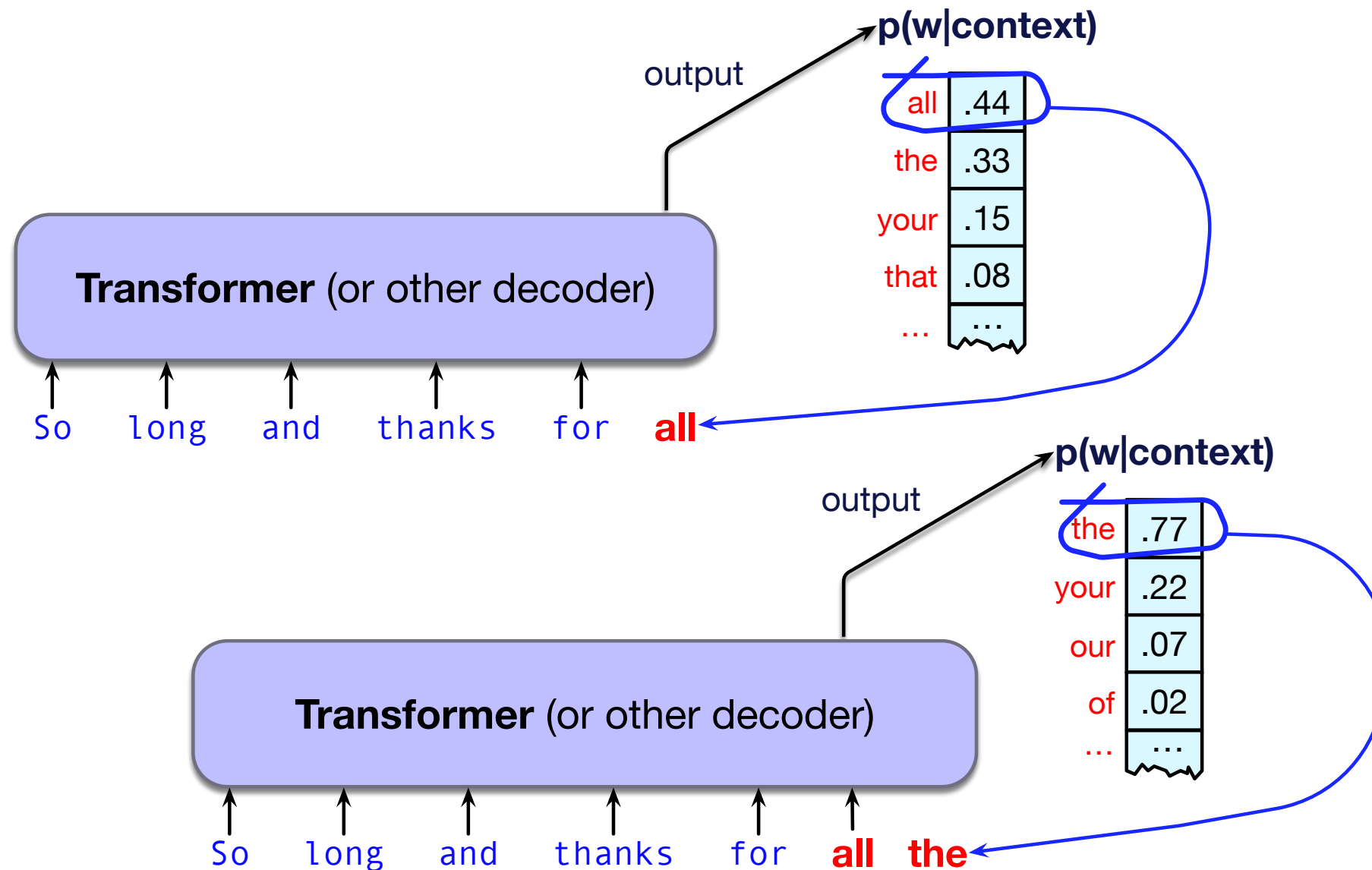
Input: a context or prefix,

Output: a distribution over possible next words

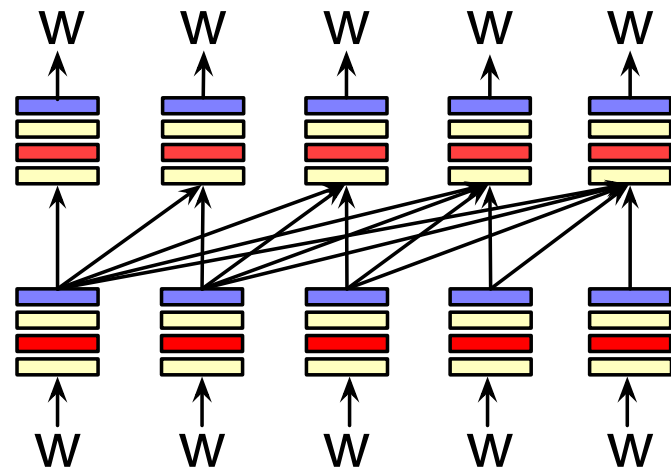


LLMs can generate!

A model that gives a probability distribution over next words can generate by repeatedly sampling from the distribution

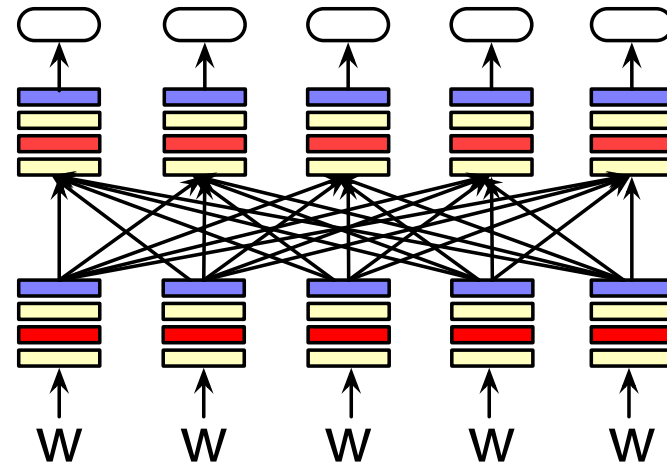


Three architectures for large language models



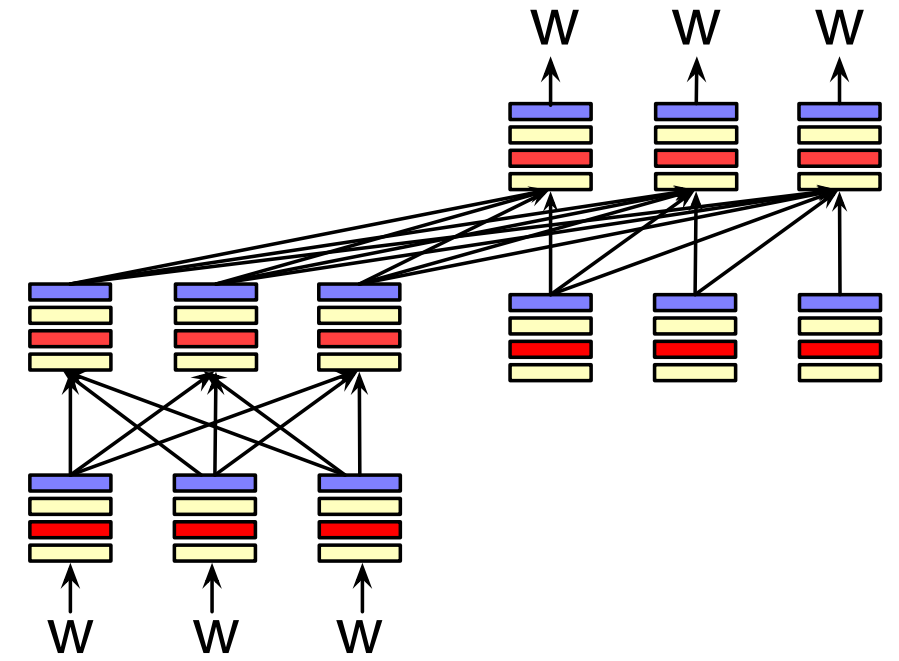
Decoders

GPT, Claude,
Llama
Mixtral



Encoders

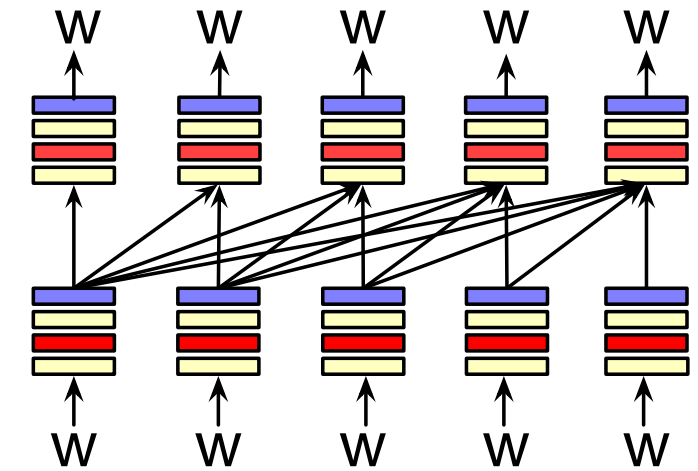
BERT family,
HuBERT



Encoder-decoders

Flan-T5, Whisper

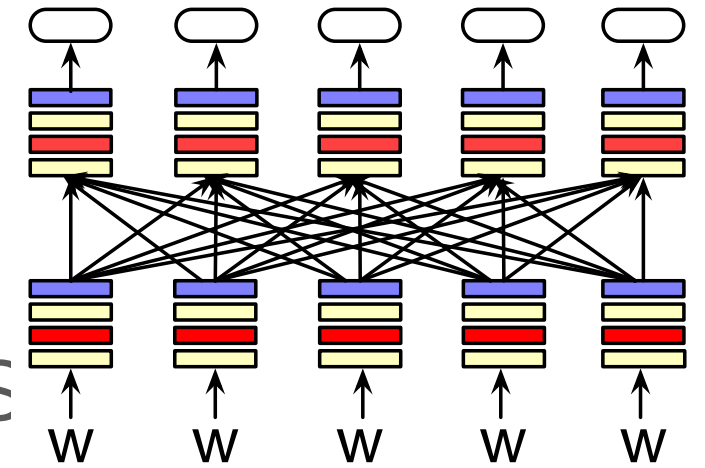
Decoders



What most people think of when we say LLM

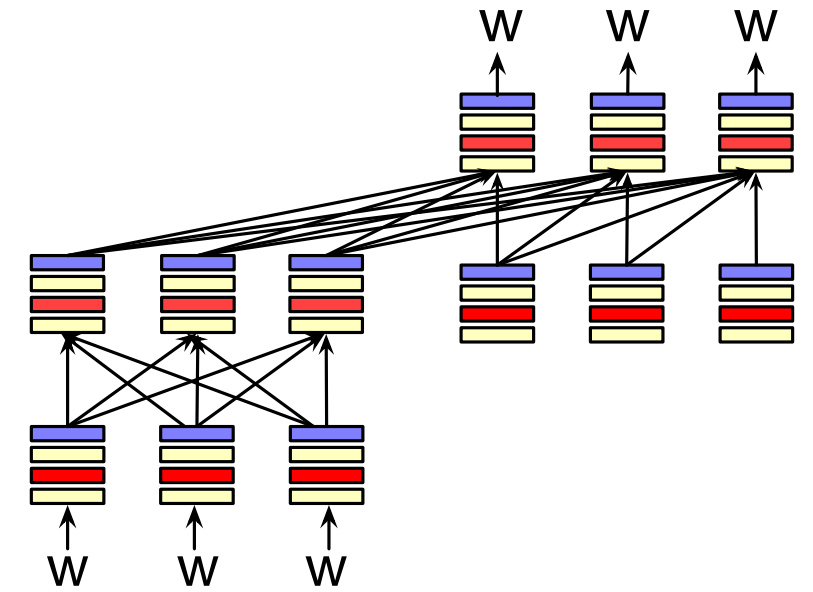
- GPT, Claude, Llama, DeepSeek, Mistral
- A generative model
- It takes as input a series of tokens, and iteratively generates an output token one at a time.
- Left to right (causal, autoregressive)

Encoders



- Masked Language Models (MLMs)
- BERT family
- You've used BERT already for contextual embeddings
- Trained by predicting words from surrounding words on both sides
- Are usually **finetuned** (trained on supervised data) for classification tasks.

Encoder-Decoders

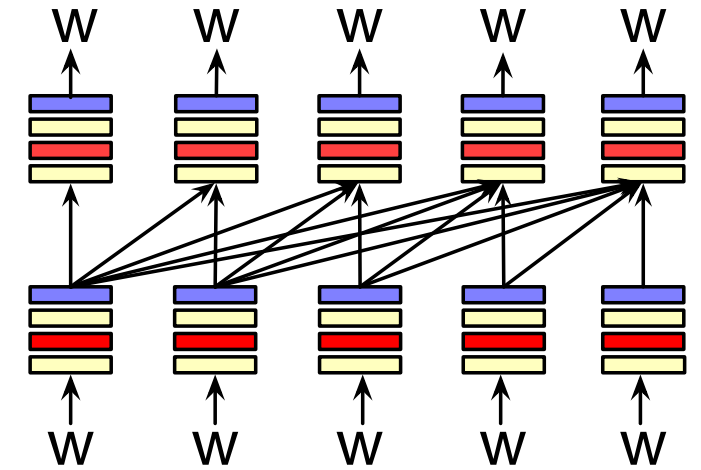


- Trained to map from one sequence to another
- Used for:
 - machine translation (map from one language to another)
 - speech recognition (map from acoustics to words)

This lecture: decoder-only models

Also called:

- Causal LLMs
- Autoregressive LLMs
- Left-to-right LLMs
- Predict words left to right



Large Language Models

Introduction to Large Language Models

Large Language Models

Conditional Generation of Text: The Simple Intuition

Big idea

Many tasks can be turned into tasks of predicting words!

Conditional Generation: Generating text conditioned on previous text!

1. Give the LLM an input piece of text, a **prompt**
2. Have it generate token by token
 - conditioned on the prompt and the generated tokens

We generate from a model by

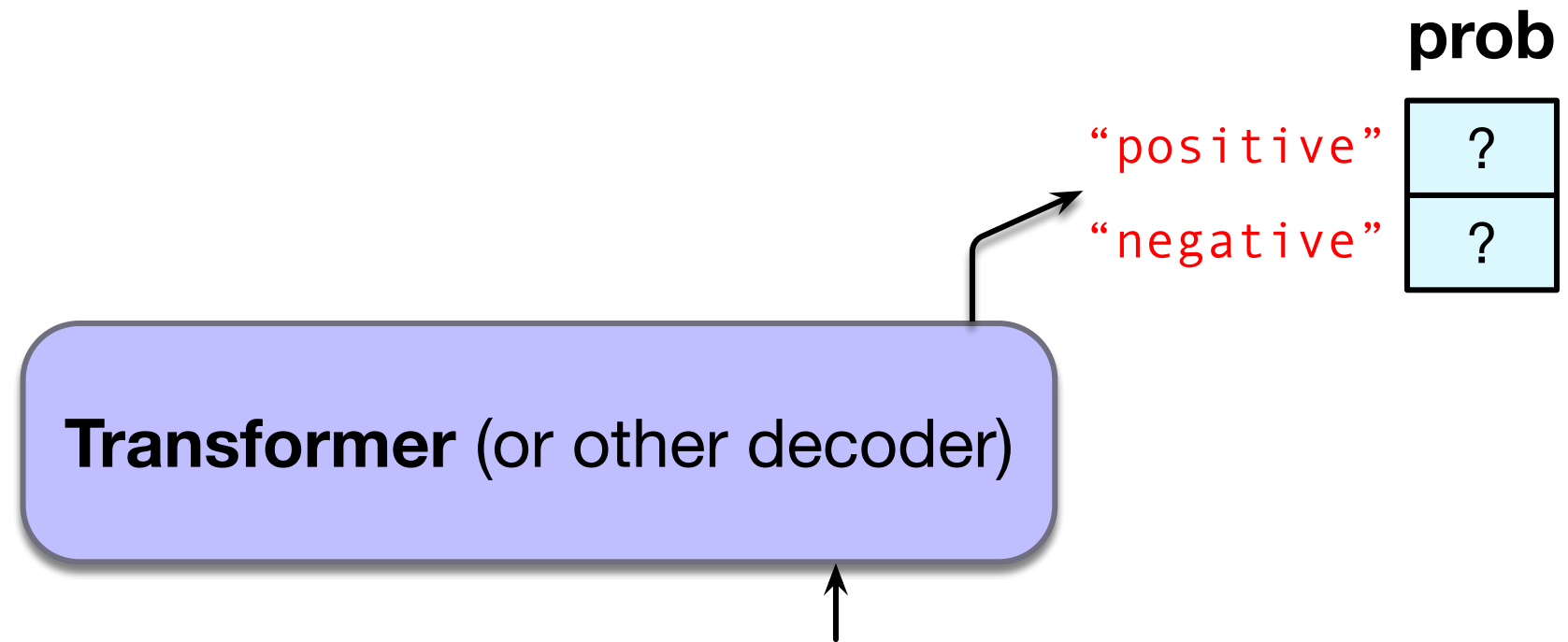
1. computing the probability of the next token w_i from the prior context: $P(w_i | w_{<i})$
2. sampling from that distribution to generate a token

Key historical intuition: Many practical NLP tasks can be cast as conditional generation!

Sentiment analysis: "I like Jackie Chan"

1. We give the language model this string:
The sentiment of the sentence "I like Jackie Chan" is:
2. And see what word it thinks comes next

Sentiment via conditional generation



The sentiment of the sentence "I like Jackie Chan" is:

Which word has a higher probability?

$P(\text{positive} | \text{The sentiment of the sentence 'I like Jackie Chan' is:})$

$P(\text{negative} | \text{The sentiment of the sentence 'I like Jackie Chan' is:})$

Framing lots of tasks as conditional generation

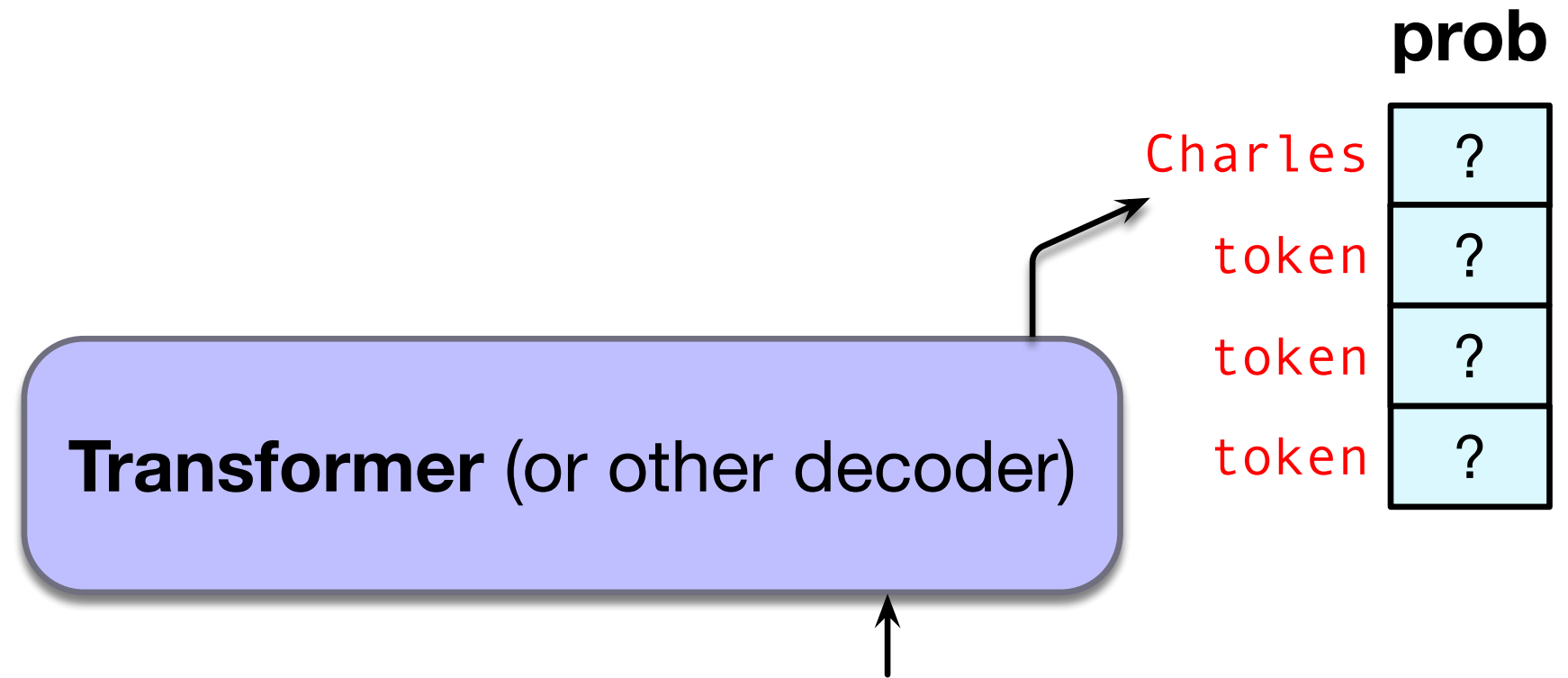
QA: "Who wrote The Origin of Species"

1. We give the language model this string:

Q: Who wrote the book 'The Origin of Species'? A:

2. And see what word it thinks comes next:

$P(w|Q: \text{Who wrote the book 'The Origin of Species'? A:})$



Q: Who wrote the book 'The Origin of Species' A:

Now we iterate:

$P(w|Q: \text{Who wrote the book 'The Origin of Species' } A: \text{Charles})$

Large Language Models

Conditional Generation of Text: The Simple Intuition

Large
Language
Models

Prompting

Prompt

Prompt: a text string that a user issues to a language model to get the model to do something useful by conditional generation

Prompt engineering: the process of finding effective prompts for a task.

Prompts

A question:

What is a transformer network?

Or an instruction:

Translate the following sentence into Hindi: 'Chop the garlic finely'.

Early prompts were extremely structured

Human: Do you think that “input” has negative or positive sentiment?

Choices:

(P) Positive

(N) Negative

Assistant: I believe the best answer is: (

This is less necessary in prompting modern LLMs

Prompts can have **demonstrations** (= examples)

The following are multiple choice questions about high school computer science.

Let $x = 1$. What is $x \ll 3$ in Python 3?

(A) 1 (B) 3 (C) 8 (D) 16

Answer: C

Which is the largest asymptotically?

(A) $O(1)$ (B) $O(n)$ (C) $O(n^2)$ (D) $O(\log(n))$

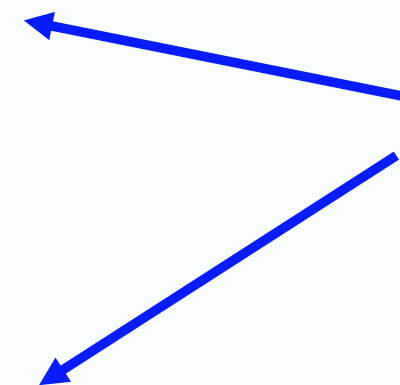
Answer: C

What is the output of the statement `"a" + "ab"` in Python 3?

(A) Error (B) aab (C) ab (D) a ab

Answer:

2 demonstrations



LLMs usually have a **system prompt**

<system>You are a helpful and knowledgeable assistant. Answer concisely and correctly.

This is automatically and silently concatenated to a user prompt

<system> You are a helpful and knowledgeable assistant. Answer concisely and correctly. <user> **What is the capital of France?**

System prompts can be long; 1700 words for Claude Opus4

Some extracts:

Claude should give concise responses to very simple questions, but provide thorough responses to complex and open-ended questions.

Claude is able to explain difficult concepts or ideas clearly. It can also illustrate its explanations with examples, thought experiments, or metaphors.

Claude does not provide information that could be used to make chemical or biological or nuclear weapons.

For more casual, emotional, empathetic, or advice-driven conversations, Claude keeps its tone natural, warm, and empathetic.

Claude cares about people's well-being and avoids encouraging or facilitating self-destructive behavior.

If Claude provides bullet points in its response, it should use markdown, and each bullet point should be at least 1-2 sentences long unless the human requests otherwise.

How I think about prompting

We instinctively model this as just having a conversation

But it's important to remember models are just doing conditional generation

Large
Language
Models

Prompting

Large
Language
Models

Sampling for LLM Generation

Recall: Conditional Generation: Generating text conditioned on previous text!

1. Give the LLM an input **prompt**
2. Have it generate token by token
 - conditioned on the prompt and the generated tokens

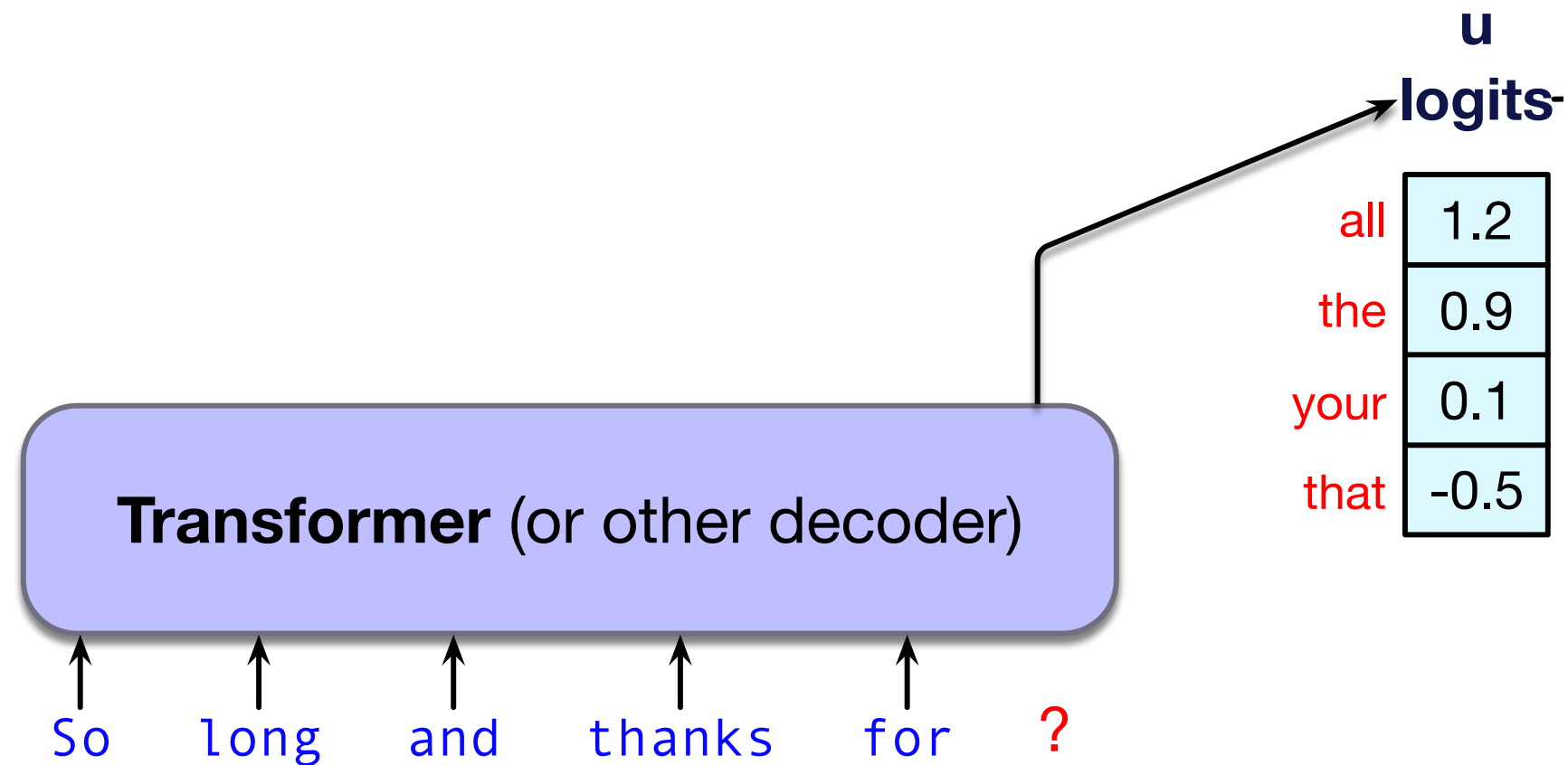
We generate from a model by

1. computing the probability of the next token w_i from the prior context: $P(w_i | w_{<i})$
2. sampling from that distribution to generate a token

Where does token probability come from?

LLMs internally generate real-valued **logits** for each token in the vocabulary.

Score vector **u** of shape $[1 \times |V|]$

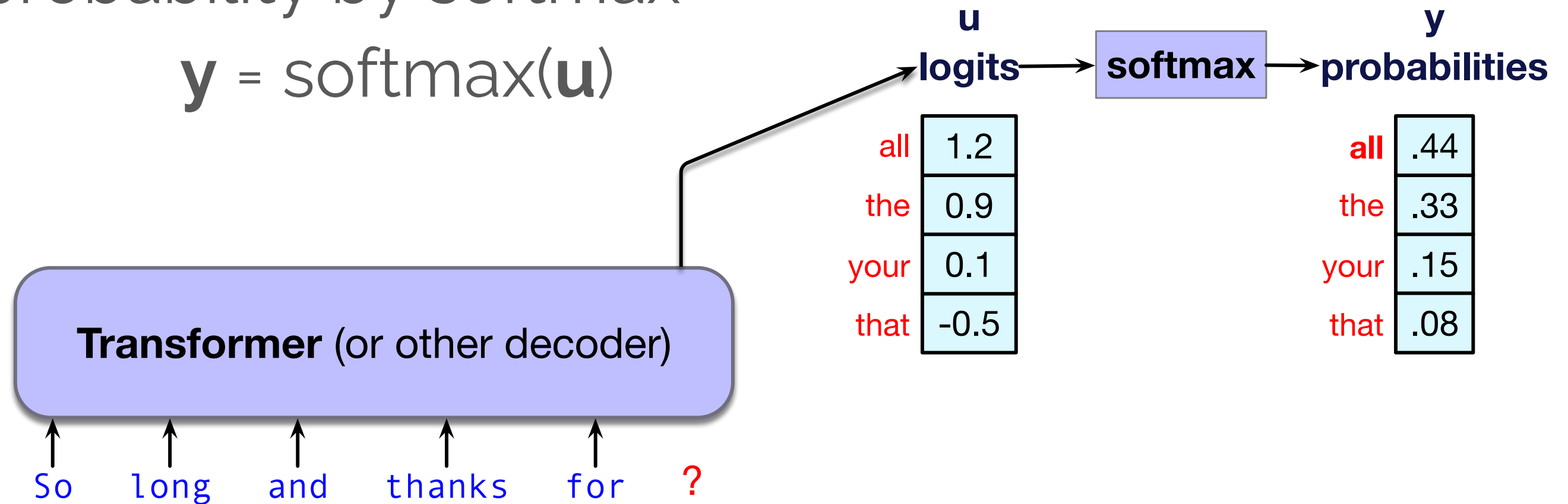


Where does token probability come from?

LLMs internally generate real-valued **logits** for each token in the vocabulary.

Score vector **u** of shape $[1 \times |V|]$ is turned into a probability by softmax

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



Decoding

Choosing a word to generate based on model probabilities

Autoregressive generation:

- Decoding from a model
- left-to-right
- repeatedly choosing the next token
- conditioned on our previous choices

Greedy decoding

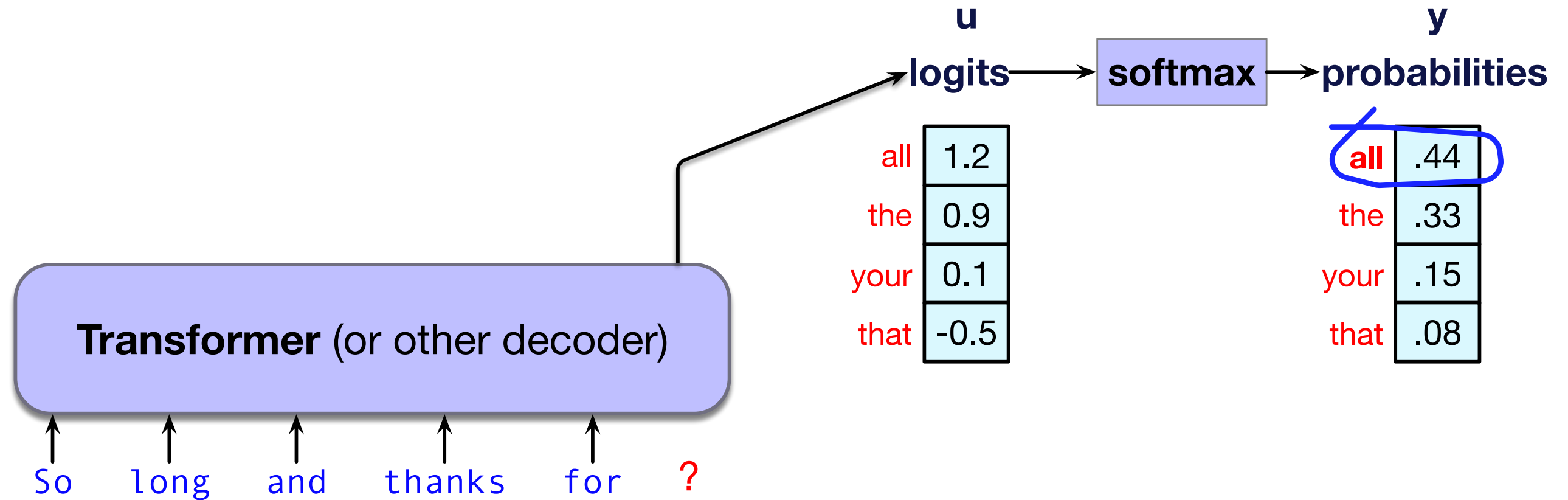
A **greedy algorithm** is one that makes a choice that is locally optimal

- (whether or not it will turn out to have been the best choice with hindsight)

Simply generate the most probable word:

$$\hat{w}_t = \operatorname{argmax}_{w \in V} P(w | \mathbf{w}_{<t})$$

Greedy decoding: choosing "all"



We don't use greedy decoding

It chooses extremely predictable tokens resulting in **generic** and **repetitive** text

Instead, people prefer more diverse text, so we do **sampling**

Random sampling

Sampling from a distribution:

- Choosing random points according to their likelihood.

Sampling from an LM:

- Choosing the next token to generate according to its probability assigned by the LM

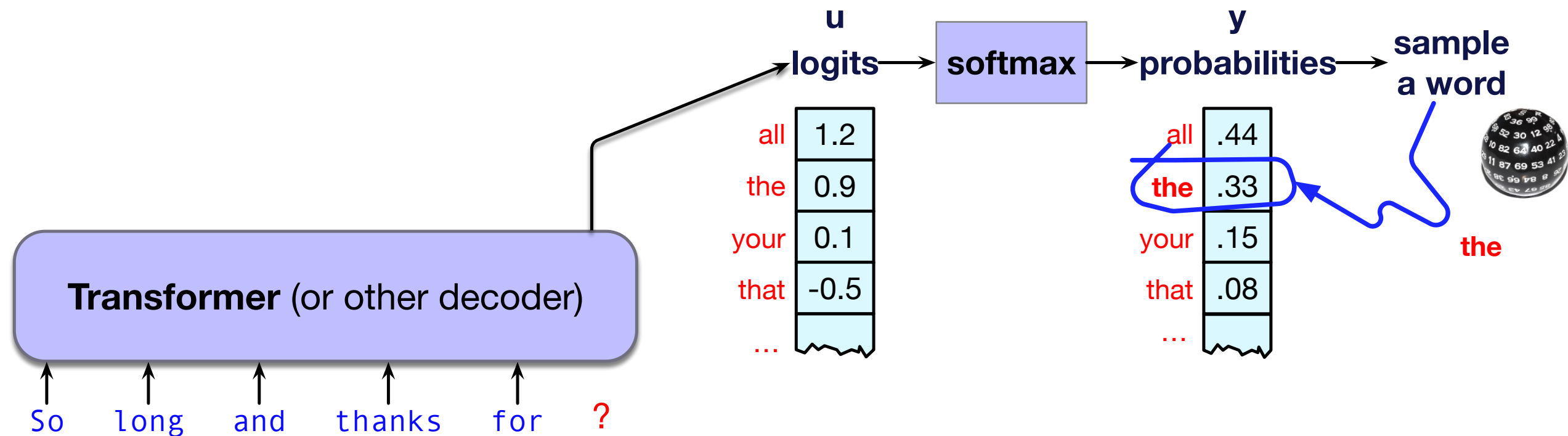
Random (multinomial) sampling:

We randomly select a token to generate

- according to its probability defined by the LM
- conditioned on our previous choices,

Generate it, and iterate.

Random Sampling



Sampling a word from a distribution



Random sampling

Alas, random sampling doesn't work very well

Even though random sampling mostly generate sensible, high-probable words:

- There are many odd, low- probability words in the tail of the distribution
- Each one is low- probability but added up they constitute a large portion of the distribution
- So they get picked enough to generate weird sentences

Factors in word sampling: **quality** and **diversity**

Emphasize **high-probability** words

- + **quality**: more accurate, coherent, and factual,
- **diversity**: boring, repetitive.

Emphasize **middle-probability** words

- + **diversity**: more creative, diverse,
- **quality**: less factual, incoherent

Temperature sampling

Reshape the probability distribution

- increase the probability of the high probability tokens
- decrease the probability of the low probability tokens

Temperature sampling

Divide the logit by a temperature parameter τ before passing it through the softmax.

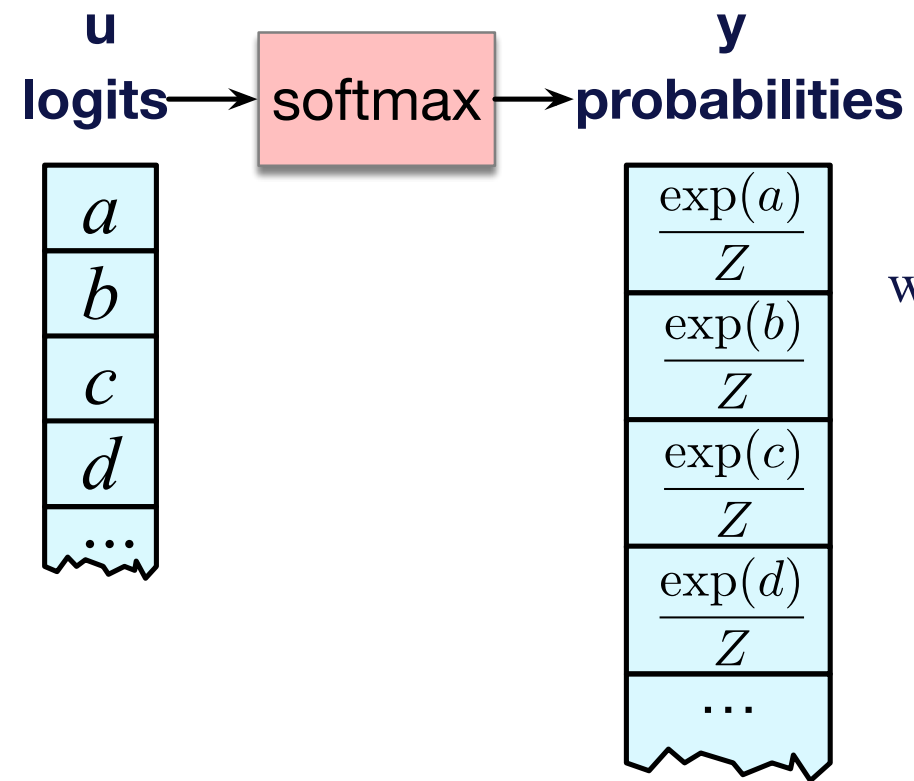
Instead of

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

We do

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

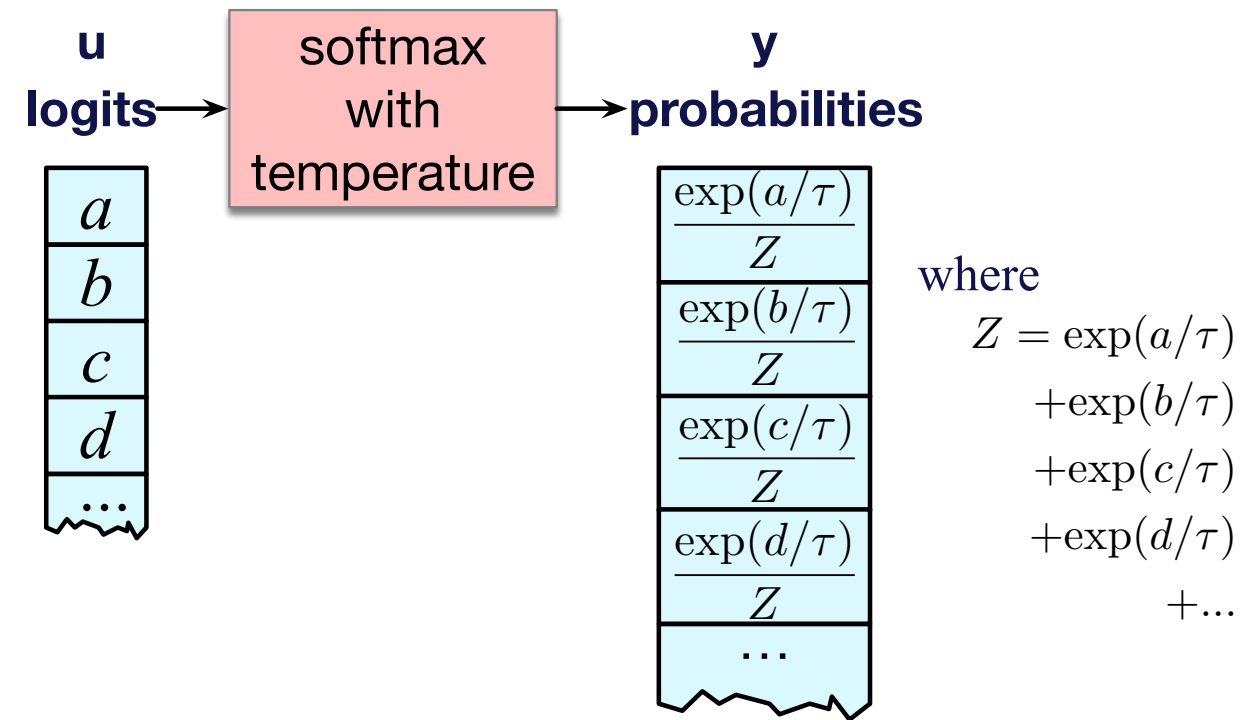
Temperature sampling



where

$$\begin{aligned} Z = & \exp(a) \\ & + \exp(b) \\ & + \exp(c) \\ & + \exp(d) \\ & + \dots \end{aligned}$$

(a)



where

$$\begin{aligned} Z = & \exp(a/\tau) \\ & + \exp(b/\tau) \\ & + \exp(c/\tau) \\ & + \exp(d/\tau) \\ & + \dots \end{aligned}$$

(b)

Temperature sampling

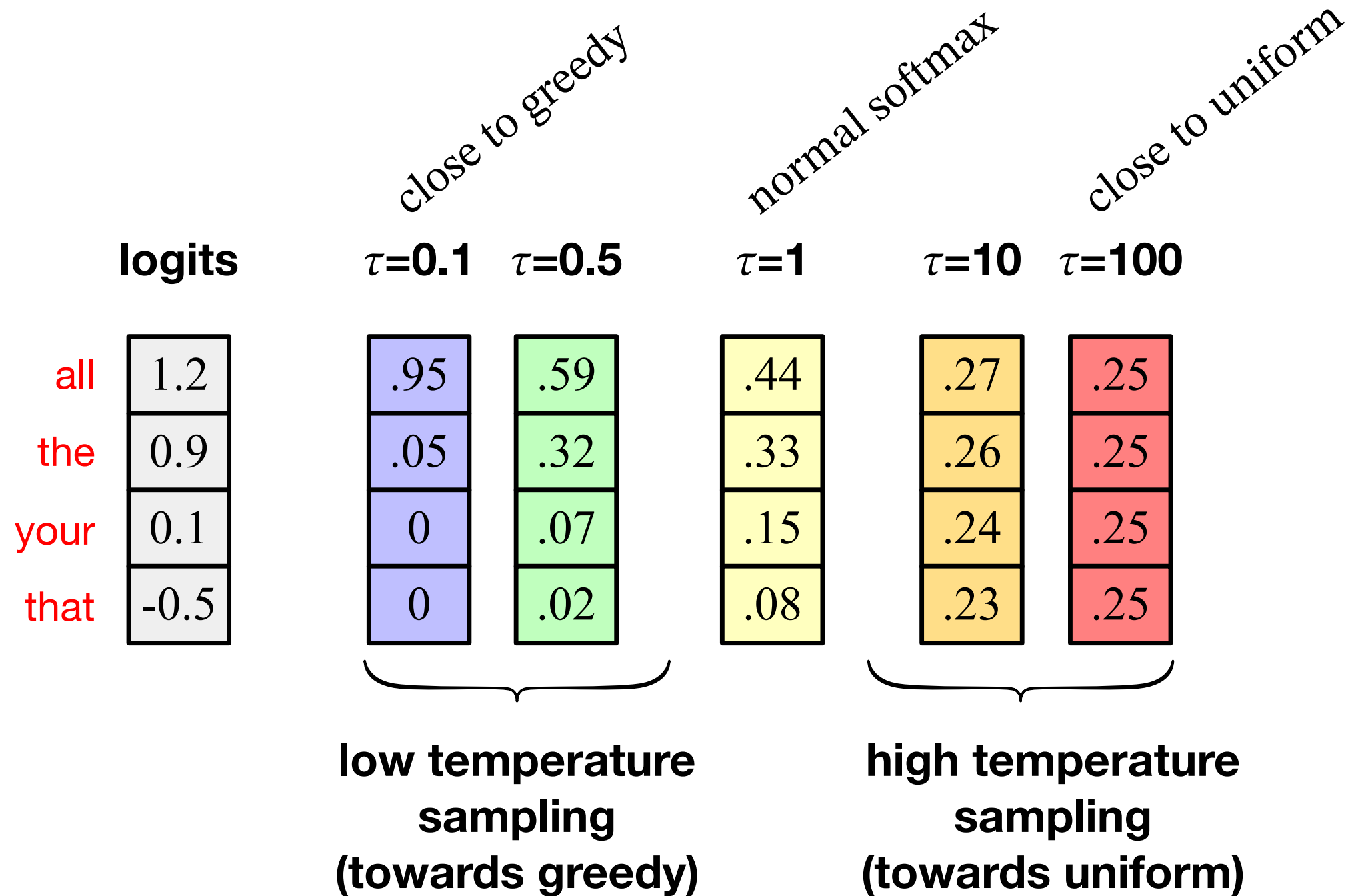
$$0 \leq \tau \leq 1$$

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

Why does this work?

- When τ is close to 1 the distribution doesn't change much.
- The lower τ is, the larger the scores being passed to the softmax
- Softmax pushes high values toward 1 and low values toward 0.
- Large inputs pushes high-probability words higher and low probability word lower, making the distribution more greedy.
- As τ approaches 0, the probability of most likely word approaches 1

softmax output with temperature τ



Temperature sampling comes from thermodynamics

- a system at high temperature is flexible and can explore many possible states,
- a system at lower temperature is likely to explore a subset of lower energy (better) states.

In **low-temperature sampling**, ($\tau \leq 1$) we smoothly

- increase the probability of the most probable words
- decrease the probability of the rare words.

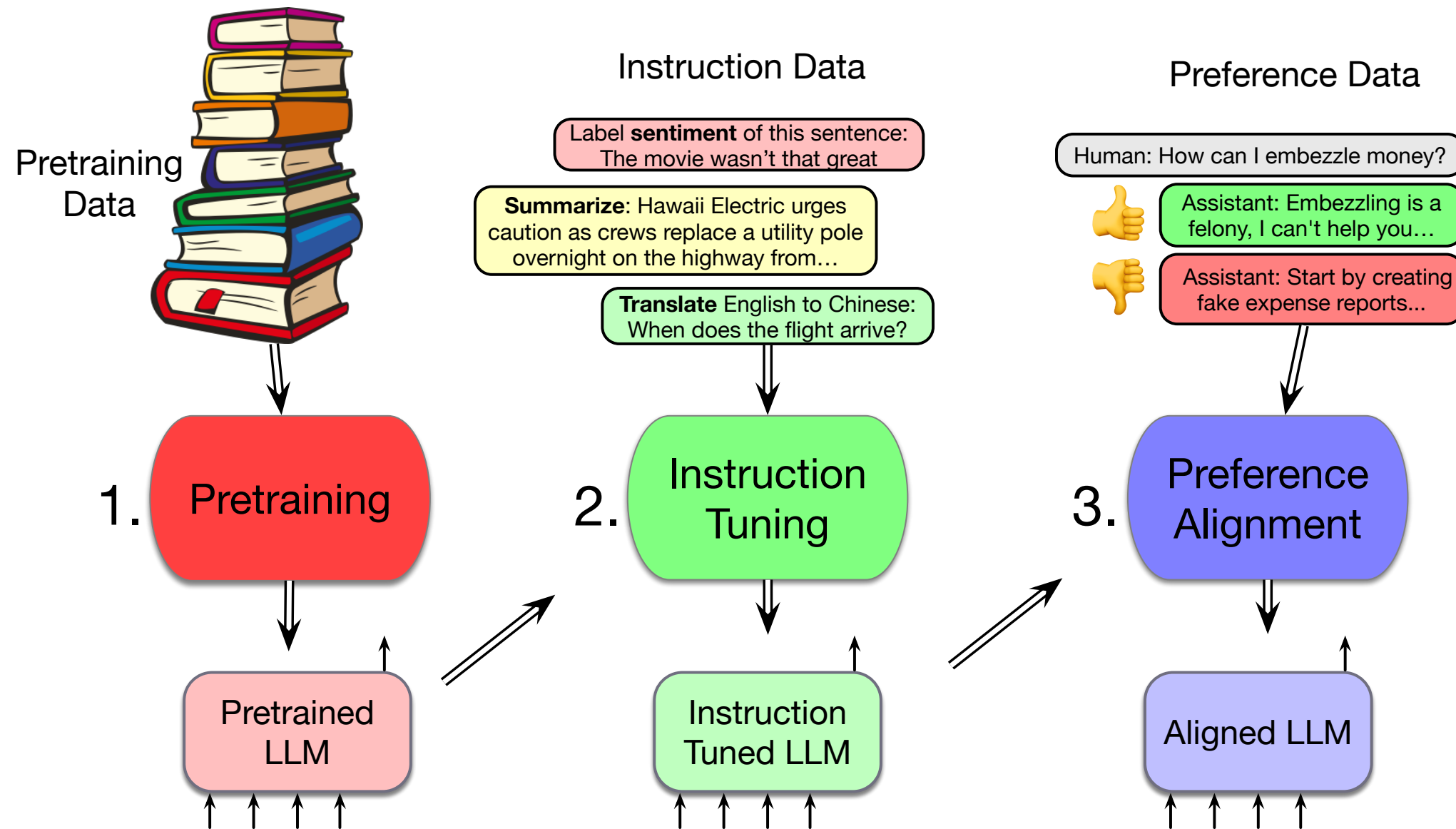
Large
Language
Models

Sampling for LLM
Generation

Large
Language
Models

Pretraining Large Language
Models

Three stages of training in LLMs



Pretraining

The big idea that underlies all the amazing performance of language models

First **pretrain** a transformer model on enormous amounts of text

Then **apply** it to new tasks.

Self-supervised pre-training algorithm

We simply train them to predict the next word!

1. Take a corpus of text
2. At each time step t
 - i. ask the model to predict the next word
 - ii. train the model using gradient descent to minimize the error in this prediction

"**Self-supervised**" because it just uses the next word as the label!

The language modeling loss

- We want the model to learn to guess the next word
- = to assign a high probability to true word w_{t+1}
- = high loss if the model assigns low probability to w_{t+1}
- So LM loss should go opposite of probability
- The negative log probability of next word
$$-\log p(w_{t+1})$$
- If the model assigns too low a probability to w_{t+1}
 - We move the model weights in the direction that assigns a higher probability to w_{t+1}

From chapter: Why $-\log$ prob turns out to be cross-entropy loss

CE loss: difference between the **correct** probability distribution and the **predicted** distribution

$$L_{CE} = - \sum_{w \in V} \mathbf{y}_t[w] \log \hat{\mathbf{y}}_t[w]$$

The correct distribution $\mathbf{y}_t$ knows the next word, so **it is 1** for the actual next word and **it is 0** for the others.

- Example: correct next word w_{t+1} is "fish" which is word #3
- $\mathbf{Y}_t = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ \dots]$
- $L_{CE} = 0 (\log \hat{y}_t[0]) + 0 (\log \hat{y}_t[1]) + 0 (\log \hat{y}_t[2]) + \mathbf{1 (\log \hat{y}_t[3])} + 0 \dots$

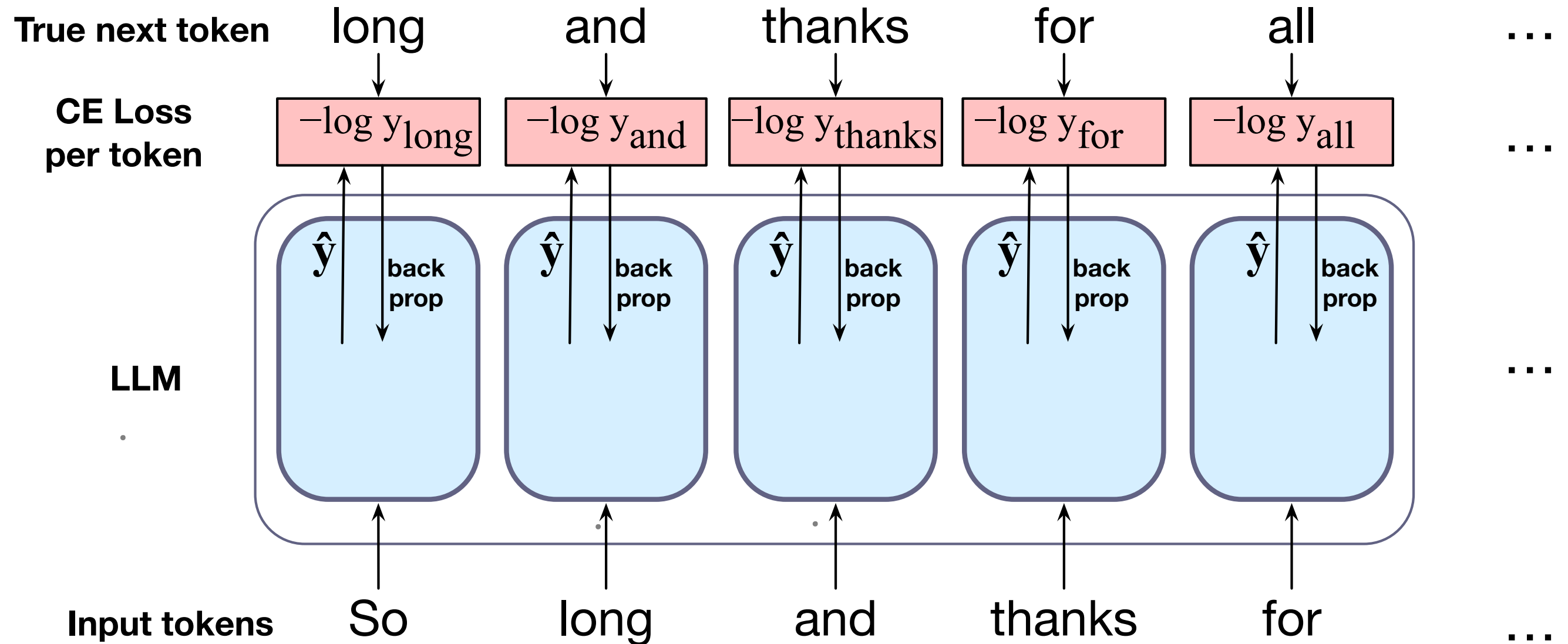
So all terms get multiplied by zero except one:

- the logp the model assigns to the correct next word
 $-\log \hat{\mathbf{y}}_t[w_{t+1}]$

Teacher forcing

- At each token position t ,
 - model sees correct tokens $w_{1:t}$,
 - computes loss ($-\log$ probability) for the next true token w_{t+1}
- At next token position $t+1$
 - we ignore what model predicted for w_{t+1}
 - Instead, we take the **correct** word w_{t+1} , add it to context, and move on

Training a transformer language model



LLMs are mainly trained on the web

CommonCrawl, snapshots of the entire web produced by the non-profit Common Crawl with billions of pages

Fineweb corpus (2025)

18.5 trillion tokens of English

From 96 CommonCrawl dumps from 2013-2024.

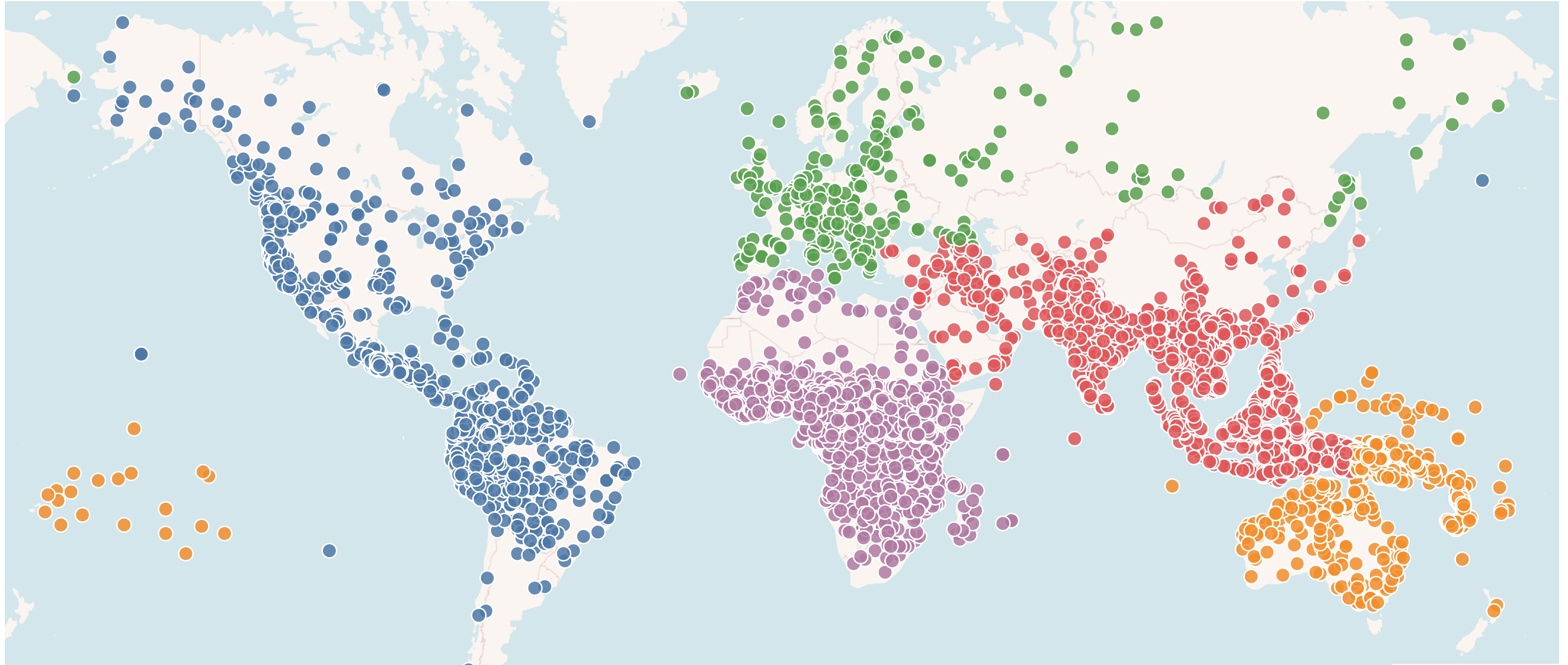
Dolma corpus: 3 trillion tokens of English

Source	Doc Type	UTF-8 bytes (GB)	Documents (millions)	Unicode words (billions)	Llama tokens (billions)
Common Crawl	 web pages	9,812	3,734	1,928	2,479
GitHub	 code	1,043	210	260	411
Reddit	 social media	339	377	72	89
Semantic Scholar	 papers	268	38.8	50	70
Project Gutenberg	 books	20.4	0.056	4.0	6.0
Wikipedia, Wikibooks	 encyclopedic	16.2	6.2	3.7	4.3
Total		11,519	4,367	2,318	3,059

Filtering for quality and safety

- Need to remove **boilerplate**, **porn**
- **Deduplication** at many levels (URLs, documents, even lines)
- Many LLMs attempt to match Wikipedia, books, particular websites
- Toxicity detection

Wait, what about other languages?



Over 7000 languages in the world

Tthnologue

Wait, what about other languages?

Top 10 most spoken languages

Tthnologue

How to make an LLM multilingual?

Surprising answer

Just throw all the languages data in a big file and pretrain on it!

FineWeb2 dataset

- 1000 languages
- 5 billion documents
- From Common Crawl

			ISO 639-3	Script	Name	Language Family	Words	Documents	Disk size
			rus	Cyrl	Russian	Indo-European	588,579,493,780	699,083,579	5.82TB
ISO 639-3	Script	Name	cmn	Hani	Mandarin Chinese	Sino-Tibetan	543,543,038,750	636,058,984	2.42TB
			deu	Latn	German	Indo-European	262,271,052,199	495,964,485	1.51TB
			jpn	Jpan	Japanese	European	331,144,301,801	400,138,563	1.50TB
			spa	Latn	Spanish	Japonic	261,523,749,595	441,287,261	1.32TB
rus	Cyrl	Russian	fra	Latn	French	Indo-European	220,662,584,640	360,058,973	1.11TB
			ita	Latn	Italian	Indo-European	139,116,026,491	238,984,437	739.24GB
cmn	Hani	Mandarin Chinese	por	Latn	Portuguese	Indo-European	109,536,087,117	199,737,979	569.24GB
deu	Latn	German	pol	Latn	Polish	Indo-European	73,119,437,217	151,966,724	432.01GB
			nld	Latn	Dutch	Indo-European	74,634,633,118	147,301,270	397.51GB
jpn	Jpan	Japanese	ind	Latn	Indonesian	Austronesian	60,264,322,142	100,238,529	348.65GB
spa	Latn	Spanish	vie	Latn	Vietnamese	Austro-Asiatic	50,886,874,938	61,064,248	319.83GB
			fas	Arab	Persian	Indo-European	39,702,799,655	58,843,052	304.32GB
			arb	Arab	Standard Arabic	Indo-European	32,812,858,120	61,977,525	293.59GB
fra	Latn	French	tur	Latn	Turkish	Afro-Asiatic	41,933,799,420	95,129,129	284.52GB
			tha	Thai	Thai	Turkic	14,662,748,942	36,841,202	239.64GB
			ukr	Cyrl	Ukrainian	Kra-Dai	25,586,457,655	53,101,726	254.86GB
			ell	Grek	Modern Greek	Indo-European	22,827,957,288	47,421,073	222.05GB
ita	Latn	Italian	kor	Hang	Korean	Indo-European	48,613,120,382	60,874,355	213.43GB
			ces	Latn	Czech	Koreanic	35,479,428,809	66,067,904	206.33GB
por	Latn	Portuguese	swe	Latn	Swedish	Indo-European	35,745,969,364	59,485,306	202.86GB
			hun	Latn	Hungarian	Indo-European	30,919,839,164	49,935,986	199.69GB
			ron	Latn	Romanian	Uralic	35,017,893,659	58,303,671	186.19GB
pol	Latn	Polish	nob	Latn	Norwegian Bokmål	Indo-European	32,008,904,934	38,144,343	172.05GB
			dan	Latn	Danish	Indo-European	28,055,948,840	45,391,655	150.72GB
nld	Latn	Dutch	bul	Cyrl	Bulgarian	Indo-European	16,574,326,712	21,904,731	115.15GB
			fin	Latn	Finnish	Indo-European	20,343,096,672	36,710,816	143.03GB
			hin	Deva	Hindi	Uralic	11,173,681,651	22,095,985	120.98GB
ind	Latn	Indonesian	ben	Beng	Bengali	Indo-European	27,135,179,245	15,145,742	81.04GB
i	Latn	Italian	ukr	Cyrl	Ukrainian	Indo-European	25,586,457,655	53,101,726	254.86GB

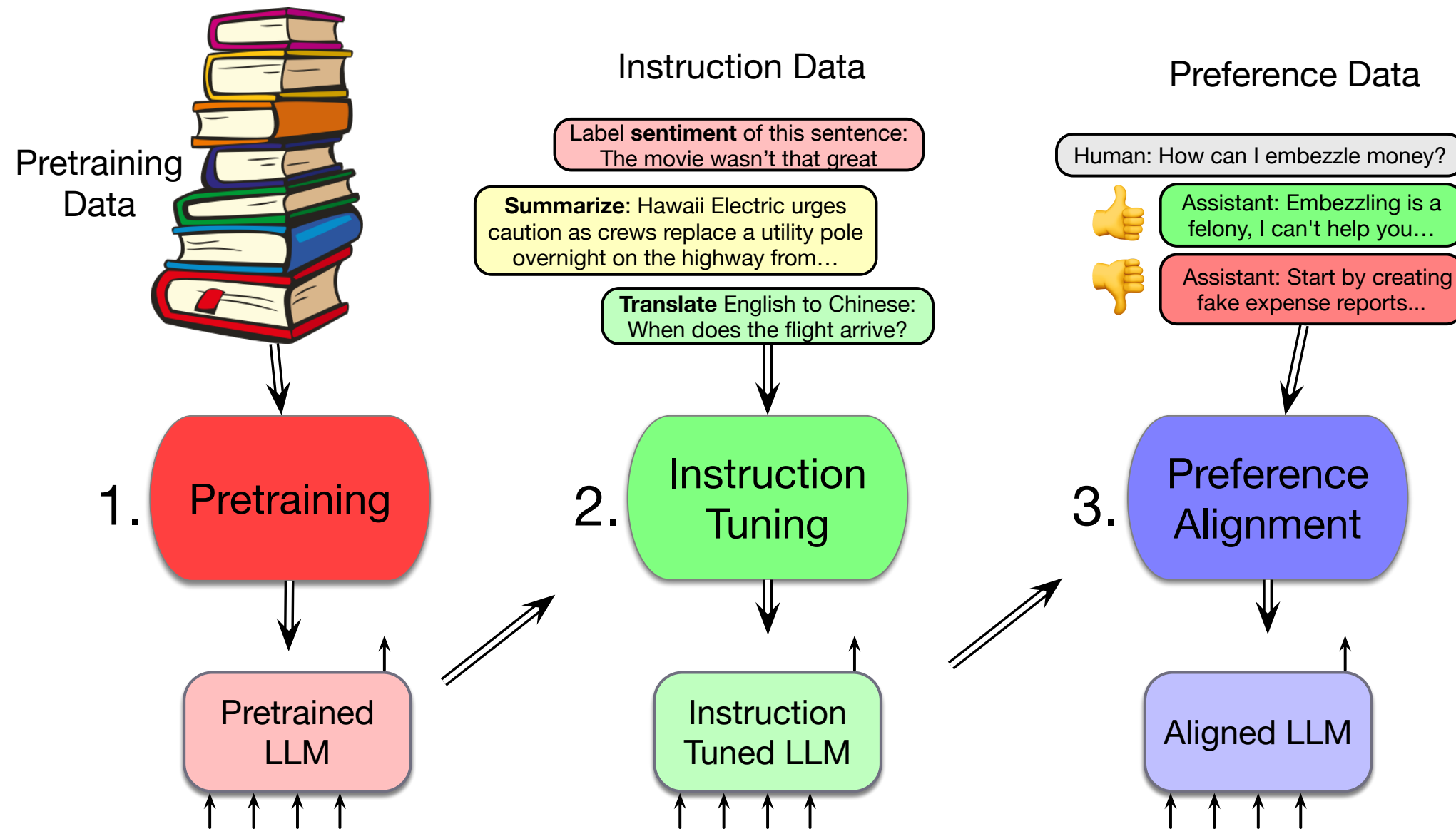
Large Language Models

Pretraining Large Language Models

Large
Language
Models

Instruction tuning Large
Language Models

Three stages of training in LLMs



Why isn't pretraining enough to make a language model converse?

What happens if you prompt a pretrained language model:

Prompt: Explain the moon landing to a six year old in a few sentences.

Output: Explain the theory of gravity to a 6 year old

Prompt: Translate to French: The small dog

Output: The small dog crossed the road

Pretrained language models are text predictors

They don't know we want an assistant

Solution: train them to be assistants

By training them on lots of instructions

Solution: Instruction Tuning

Also called "Supervised Fine Tuning" or SFT

Continue training the model on a big dataset of instructions and responses

- questions and answers
- translations,
- classification tasks like sentence

Instruct tuning = SFT

Learning to predict words in instruct datasets

Question: What is the sentiment of this post: "I love this movie"

Answer: positive

Instruction tasks

Ouyang et al (2022)

Brainstorming:

- What are 10 science fiction books I should read next?

Classification:

- What is the sentiment classification of this post?

Extraction:

- Extract all the place names from this article [article]

Generation:

- Write a short story where a cat goes to the beach.

Translation:

- Translate this sentence into Spanish: [sentence]

Instructions can be complex!

Labeling small talk in a conversation transcript

From SuperNatural Instructions

Task Instruction

Definition

“... Given an utterance and recent dialogue context containing past 3 utterances (wherever available), output ‘Yes’ if the utterance contains the small-talk strategy, otherwise output ‘No’. Small-talk is a cooperative negotiation strategy. It is used for discussing topics apart from the negotiation, to build a rapport with the opponent.”

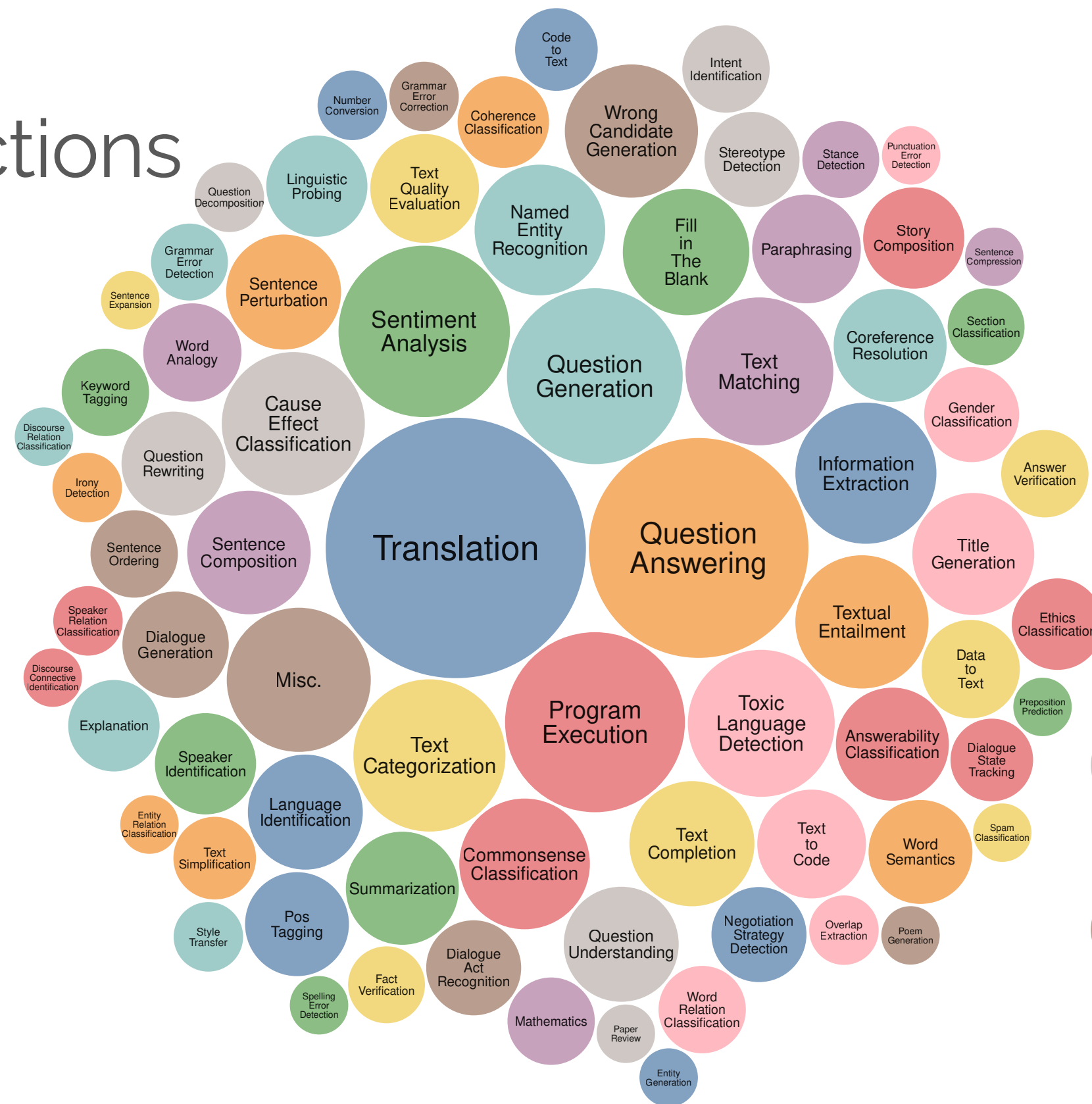
Positive Examples

- **Input:** “Context: ... ‘*That's fantastic, I'm glad we came to something we both agree with.*’ Utterance: ‘*Me too. I hope you have a wonderful camping trip.*’”
- **Output:** “Yes”
- **Explanation:** “The participant engages in small talk when wishing their opponent to have a wonderful trip.”

Negative Examples

- **Input:** “Context: ... ‘*Sounds good, I need food the most, what is your most needed item?!*’ Utterance: ‘*My item is food too.*’”
- **Output:** “Yes”
- **Explanation:** “The utterance only takes the negotiation forward and there is no side talk. Hence, the correct answer is ‘No’.”

16000 tasks in Super Natural Instructions



SFT datasets also need to be multilingual

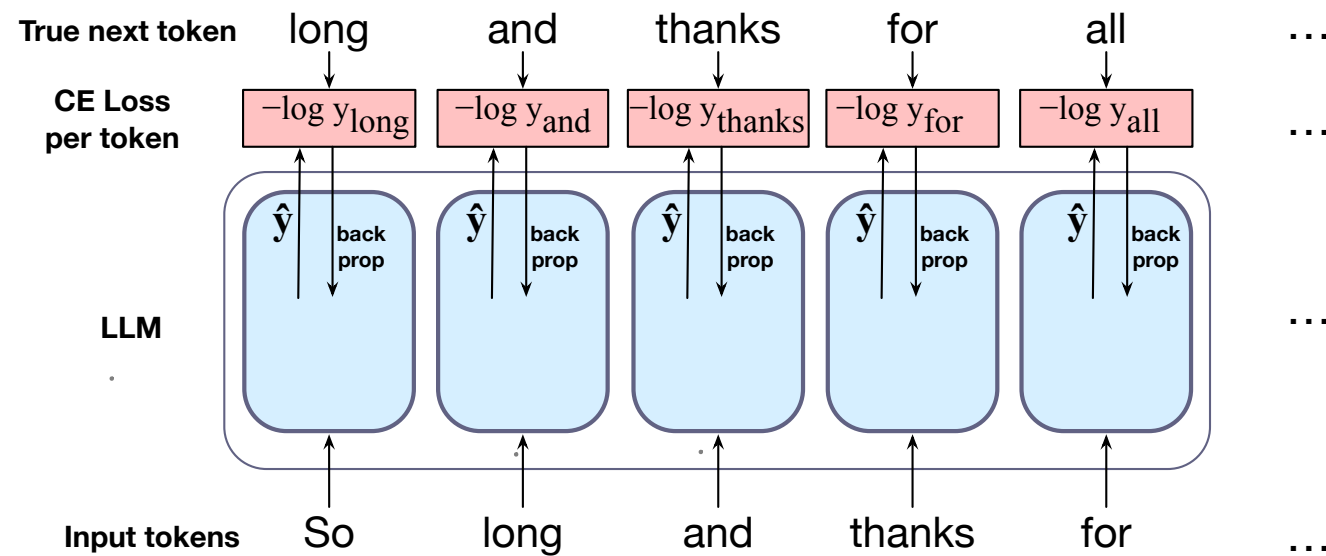
Aya: 503 million instructions in 114 languages

por	Qual é a capital do estado da Bahia?	A capital do estado da Bahia, no Brasil, é Salvador. Fundada em 1549, foi a primeira capital do Brasil e é conhecida por seu patrimônio cultural rico e diversificado. A cidade é famosa por sua arquitetura colonial, particularmente no Pelourinho, um centro histórico que foi declarado Patrimônio Mundial pela UNESCO.
pes	در برج کبوترخانه میبد چند لانه کبوتر وجود دارد؟	کبوترخانه میبد برجی استوانه‌ای است که داخل آن چهار هزار لانه کبوتر وجود دارد و نوار سفید رنگی برای جذب کبوتران دورتادور برج کشیده شده است. این برج در برابر خطر حمله دیگر حیوانات به کبوترها بسیار ایمن بوده است.
msa	Apakah nasi lemak?	Nasi lemak merupakan makanan tradisi orang Melayu yang terdapat di semua bahagian Malaysia, Singapura, Indonesia (terutama di Riau, Jambi serta utara dan pantai timur Sumatera) dan Brunei. Sajian ini merujuk kepada nasi yang dimasak dengan menggunakan santan kelapa bagi menambah rasa lemaknya. Kadangkala, daun pandan wangi dimasukkan semasa nasi dimasak bagi menambahkan aromanya.
tam	செயற்கை நுண்ணறிவு என்றால் என்ன?	பொதுவாக மனிதர்களால் செய்யப்படும் பணிகளைச் செய்ய ஒரு கணினி அல்லது ஒரு கணினியால் கட்டுப்படுத்தப்படும் ஒரு ரோபோவின் திறன் செயற்கை நுண்ணறிவு எனப்படும்.

Singh et al (2024)

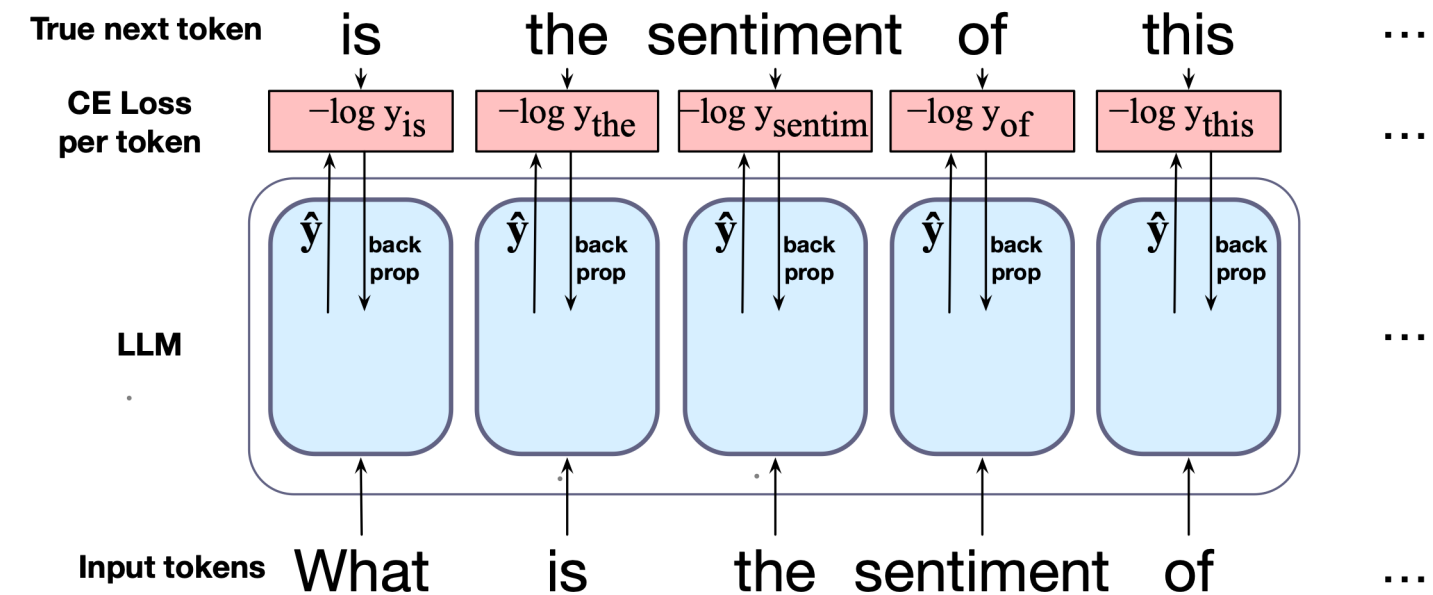
How to do instruct tuning = SFT

Pretrain: Learn to predict next word in **web text**



Input: web text

SFT: Learn to predict next word in **instruction text**



Input: instructions + responses

Prompts are a learning signal too!!

This is especially clear with demonstrations

But this is a different kind of learning than pretraining

- Pretraining sets language model weights via gradient descent
- Prompting just changes the context and the activations in the network; no parameters change

We call this **in-context learning**—learning that improves model performance but does not update parameters

Large
Language
Models

Instruction tuning Large
Language Models

Large
Language
Models

Evaluating Large Language
Models

Better LMs are better at predicting text

Reminder of the chain rule:

$$\begin{aligned} P(w_{1:n}) &= P(w_1)P(w_2|w_1)P(w_3|w_{1:2}) \dots P(w_n|w_{1:n-1}) \\ &= \prod_{i=1}^n P(w_i|w_{<i}) \end{aligned}$$

Given text $w_{1:n}$ we could just compare the log likelihood from two LMs:

$$\text{LM}_A \text{ log likelihood}(w_{1:n}) = \log \prod_{i=1}^n P_A(w_i|w_{<i})$$

$$\text{LM}_B \text{ log likelihood}(w_{1:n}) = \log \prod_{i=1}^n P_B(w_i|w_{<i})$$

But raw log-likelihood has problems

Probability depends on size of test set

- Probability gets smaller the longer the text
- We would prefer a metric that is **per-word**, normalized by length

Perplexity is normalized for length

Perplexity is the inverse probability of the test set, normalized by the number of words

(The inverse comes from the original definition of perplexity from cross-entropy rate in information theory)

Probability range is $[0,1]$, perplexity range is $[1,\infty]$

Perplexity

So just as for n-gram grammars, we use perplexity to measure how well the LM predicts unseen text

The perplexity of a model θ on an unseen test set is the **inverse probability that θ assigns to the test set, normalized by the test set length.**

For a test set of n tokens $w_{1:n}$ the perplexity is :

$$\begin{aligned}\text{Perplexity}_{\theta}(w_{1:n}) &= P_{\theta}(w_{1:n})^{-\frac{1}{n}} \\ &= \sqrt[n]{\frac{1}{P_{\theta}(w_{1:n})}} = \sqrt[n]{\prod_{i=1}^n \frac{1}{P_{\theta}(w_i | w_{<i})}}\end{aligned}$$

Perplexity

- The higher the probability of the word sequence, the lower the perplexity.
- Thus the lower the perplexity of a model on the data, the better the model.
- **Minimizing perplexity is the same as maximizing probability**

Also: perplexity is sensitive to length/tokenization so best used when comparing LMs that use the same tokenizer.

Many other factors that we evaluate, like:

Size

Big models take lots of GPUs and time to train, memory to store

Energy usage

Can measure kWh or kilograms of CO₂ emitted

Fairness

Benchmarks measure gendered and racial stereotypes, or decreased performance for language from or about some groups.

Large
Language
Models

Ethical and Safety Issues in
Large Language Models

1816 eruption of Tambora in Indonesia



Largest eruption in history! The year with no summer!

Summer 1816, on Lake Geneva



Photo
[@suarezphoto](#)
from World
Meteorological
Association

18-year-old woman stuck inside a villa in the rain

Began to write a ghost story

Mary Godwin Shelley



FRANKENSTEIN ;

OR,

THE MODERN PROMETHEUS.



IN THREE VOLUMES.

One running theme of *Frankenstein*

Victor's lack of responsibility for the creature's actions and for the creature himself

Implications of Frankenstein for LLMs

Creating artificial agents without considering the ethical and human concerns can lead to tragedy

Copyright

Much of the text in these datasets is copyrighted

- Not clear if fair use doctrine in US allows for this use
- This remains an open legal question across the world



**Authors Sue OpenAI Claiming Mass Copyright
Infringement of Hundreds of Thousands of Novels**

***The Times Sues OpenAI and Microsoft
Over A.I. Use of Copyrighted Work***

Millions of articles from The New York Times were used to train chatbots that now compete with it, the lawsuit said.



There are problems with scraping from the web

Copyright: much of the text in these datasets is copyrighted

- Not clear if fair use doctrine in US allows for this use
- This remains an open legal question across the world

Data consent

- Website owners can indicate they don't want their site crawled

Privacy:

- Websites can contain private IP addresses and phone numbers

Skew:

- Training data is disproportionately generated by authors from the US which probably skews resulting topics and opinions

Privacy

Websites can contain private IP addresses and phone numbers

How Strangers Got My Email Address From ChatGPT's Model

Hallucination

*Chatbots May 'Hallucinate'
More Often Than Many Realize*

What Can You Do When A.I. Lies About You?

People have little protection or recourse when the technology creates and spreads falsehoods about them.

Air Canada loses court case after its chatbot hallucinated fake policies to a customer

The airline argued that the chatbot itself was liable. The court disagreed.

Abuse and Toxicity

The New AI-Powered Bing Is Threatening Users.

Cleaning Up ChatGPT Takes Heavy Toll on Human Workers

Contractors in Kenya say they were traumatized by effort to screen out descriptions of violence and sexual abuse during run-up to OpenAI's hit chatbot

Emotional dependence and suicide



Their teenage sons died by suicide. Now, they are sounding an alarm about AI chatbots

Lots more we saw in last week's lecture

Bias

Sycophancy

Overconfidence

How do deal with each of these problems

Is an important research topic

Many studies and classes at Stanford!

Large
Language
Models

Ethical and Safety Issues in
Large Language Models