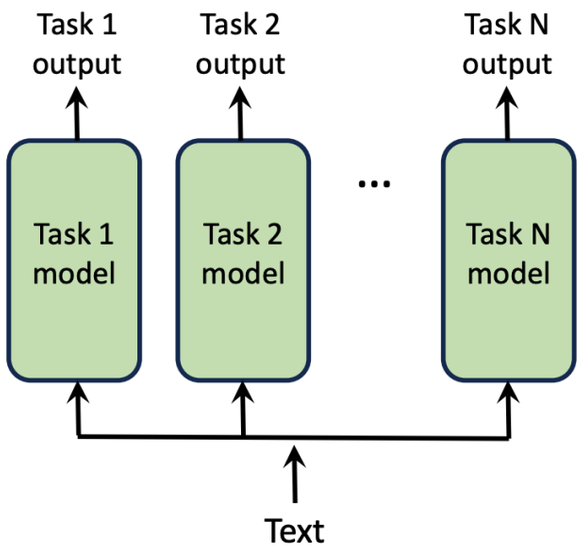
LLM Recipe: Pretraining

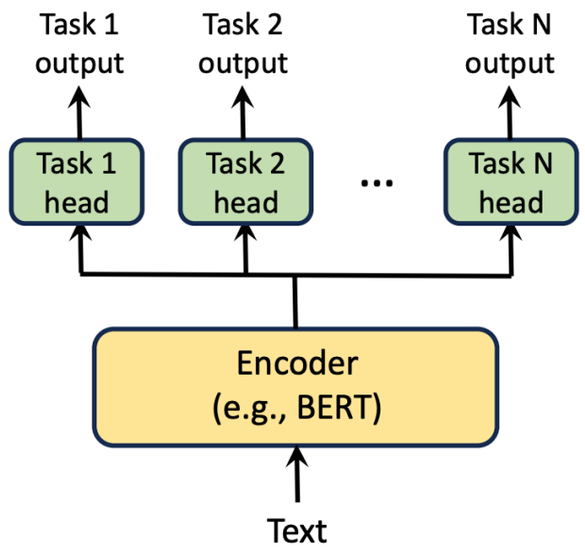
EECS 183/283a: Natural Language Processing

Evolution of NLP

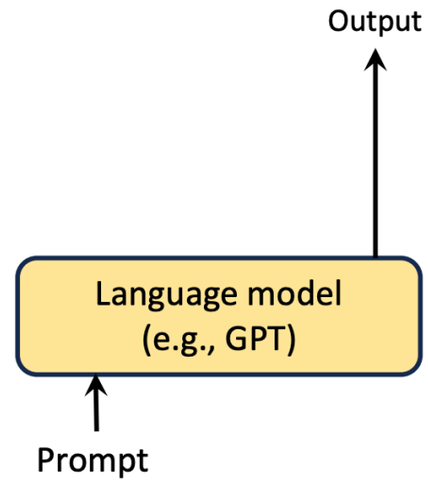
The task-specific model era (- 2018)



The encoder era (2018 - 2022)



The large language model era (2022 -)



More task-universality, less human effort

Pretraining Before LLMs

- Let's say we want to create a language technology for a new task

- We don't have a ton of data available on the new task

- But, we have a lot of general-domain language data available outside of the task (books, newspapers, etc.)

- **We can take advantage of general-domain language data by computing and using "pretrained" representations**

Why is Pretraining Helpful?

- Initializing model parameters to start in a “good” place makes learning easier for downstream tasks
  - SGD will stick relatively close to pretrained parameters
  - Gradients of finetuning loss near pretrained parameters propagate nicely
- We don’t necessarily have enough data available to train on the downstream task alone
  - SQuAD 2.0 (Rajpukar et al. 2018): <50M tokens
  - DataComp-LM (Li et al. 2024): ~240T tokens
  - Estimated available web text (Villalobos et al. 2024): ~3.1 Quadrillion tokens!

Pretrained Word Embeddings

- One popular pre-LLM approach
- E.g., Skip-Gram, GloVe — just download and initialize your favorite model with these embeddings
- Then train the model for your target task
- If you want, fine-tune the word embeddings, or keep them frozen

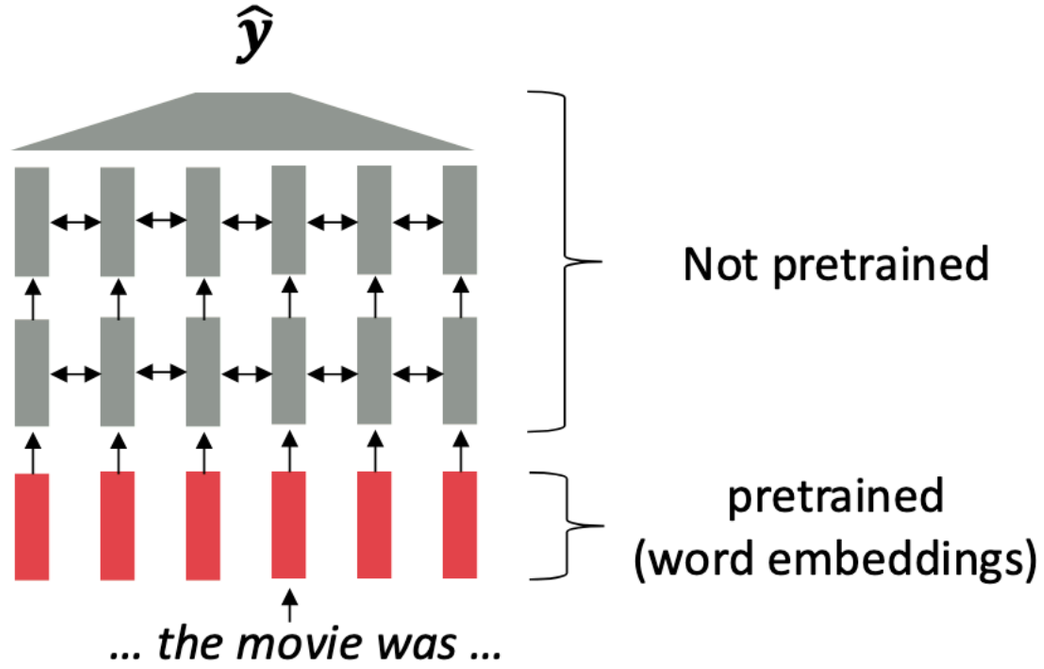


Figure from Stanford CS224n

Pretrained Language Models

- Another approach: train an entire language model on general-domain language data
- Training objective: masked language modeling or autoregressive
- Not actually used all that much prior to the popularity of models like BERT, etc.
- What do we learn that we can't learn from word embeddings alone?

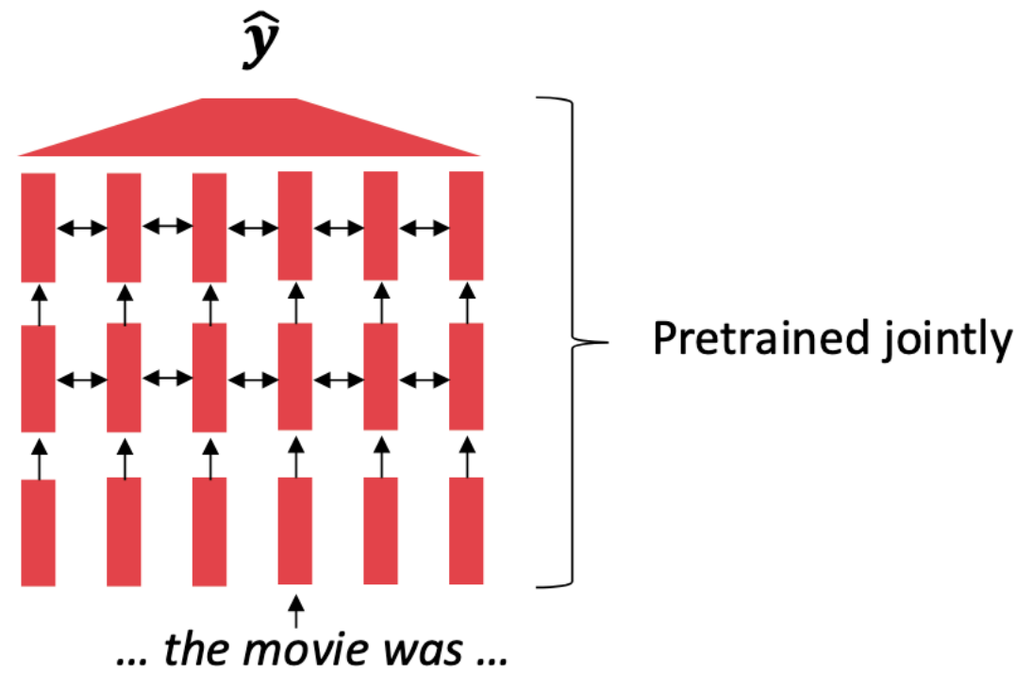


Figure from Stanford CS224n

What we Learn from Language Modeling

Cal is in _____, California

I put _____ fork down on the table

The woman walked across the street, checking for traffic over _____ shoulder

I went to the ocean to see the fish, turtles, seals, and _____

Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was ____

I'm thinking about a sequence that goes 1, 2, 3, 5, 8, 13, 21, _____

What we Learn from Language Modeling (Text)

knowledge

Cal is in _____, California

I put _____ fork down on the table

syntax

The woman walked across the street, checking for traffic over _____ shoulder

coreference

I went to the ocean to see the fish, turtles, seals, and _____

lexical semantics

Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was ____

sentiment

reasoning

I'm thinking about a sequence that goes 1, 2, 3, 5, 8, 13, 21, _____

LLM Pretraining in Practice

1. Collect some data

2. Preprocess it (filtering, tokenization, etc.)

3. Apply the relevant pretraining objective (autoregressive, masked language modeling, etc.)

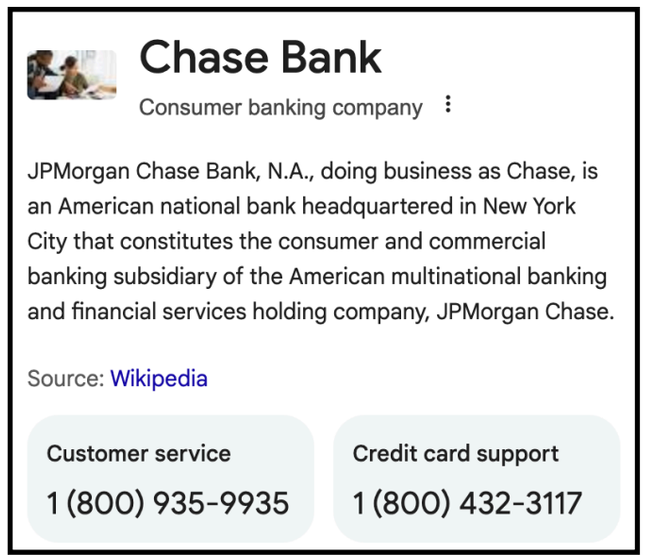
Pretraining Data

- Books
- News articles
- Web-scraped data
  - Seed a webcrawler with initial URLs
  - Download HTML from each URL
    - Scrape it for raw text
    - Scrape it for outlinks to new URLs to crawl

Data Preprocessing: Filtering

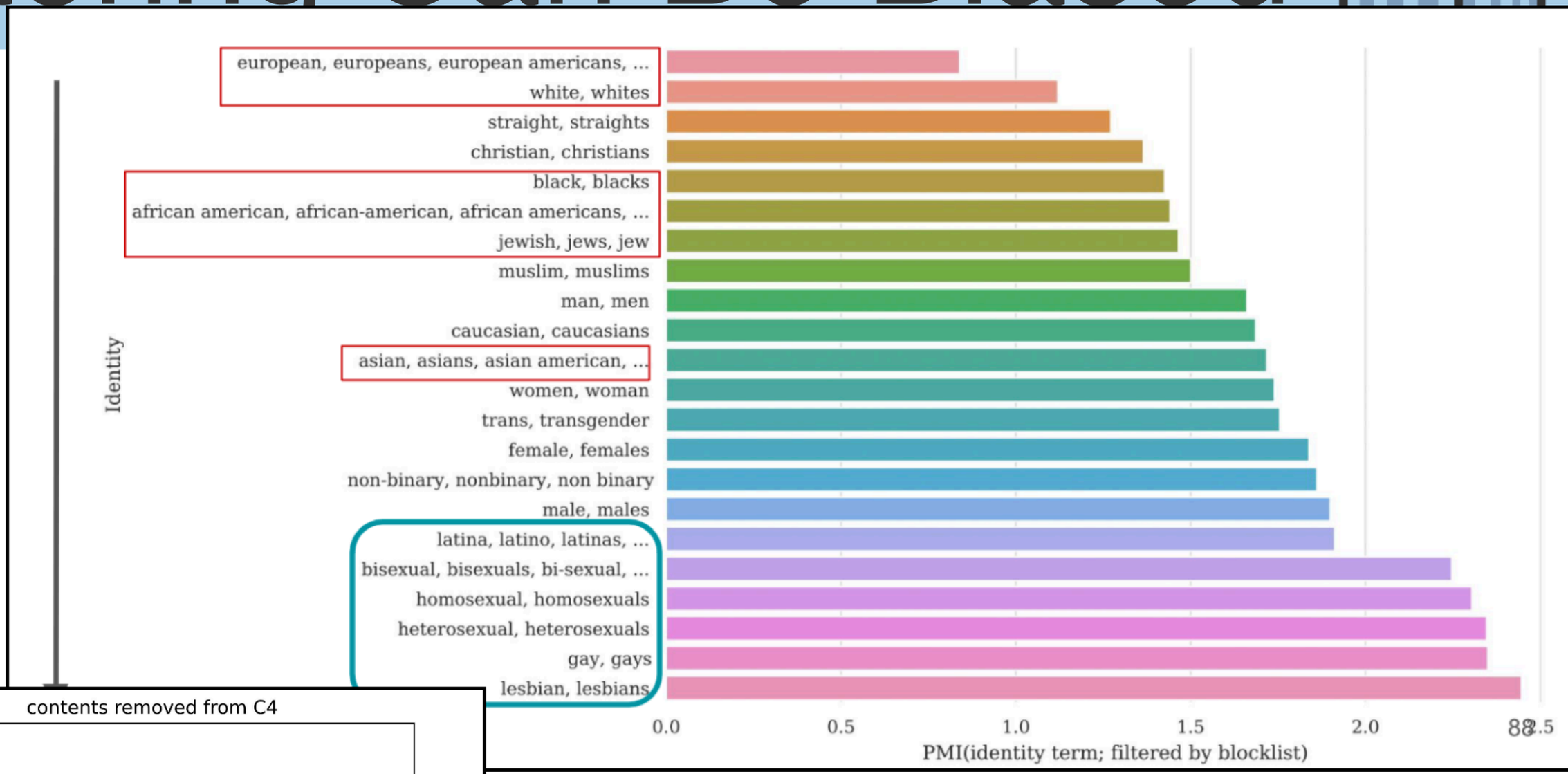
- Remove duplicate documents
- Remove junk
  - Text that is very unlikely according to a simple n-gram model
- Remove pages that are uninteresting
  - Pages that have few in-links
- Remove non-English data using a language classifier

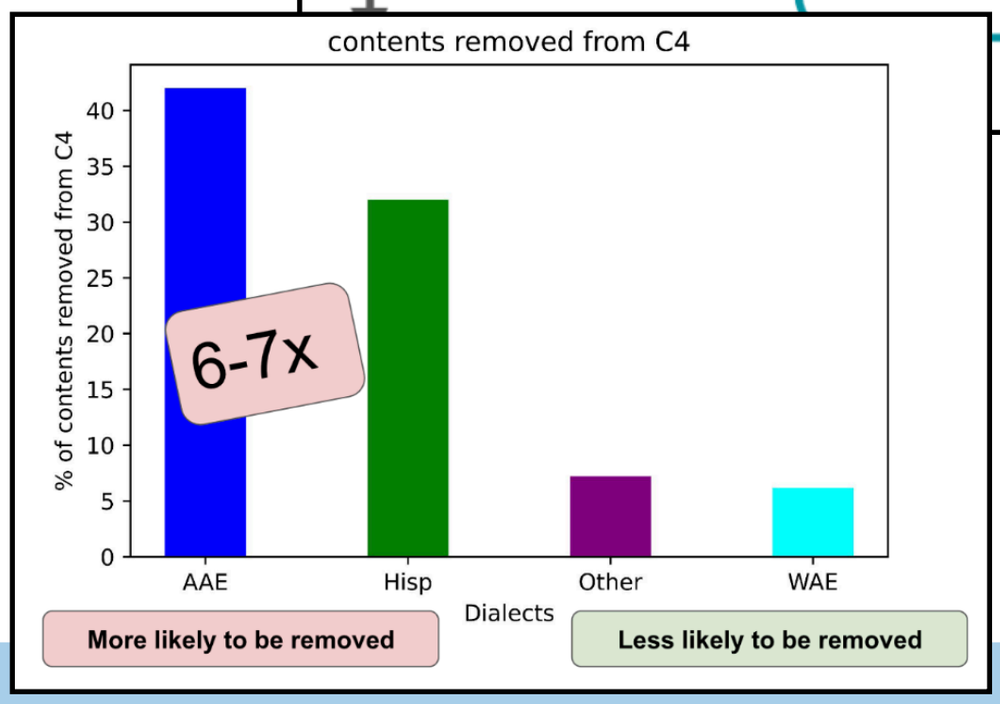
Data Preprocessing: Filtering



- Stuff we probably don't want to train on:
  - Personally identifiable information
  - Adult content → Very culturally dependent
  - Hate speech → Very culturally dependent
  - Copyrighted data → How to identify? Is it fair use?
  - NLP benchmarks (why?)

Filtering Can Be Biased



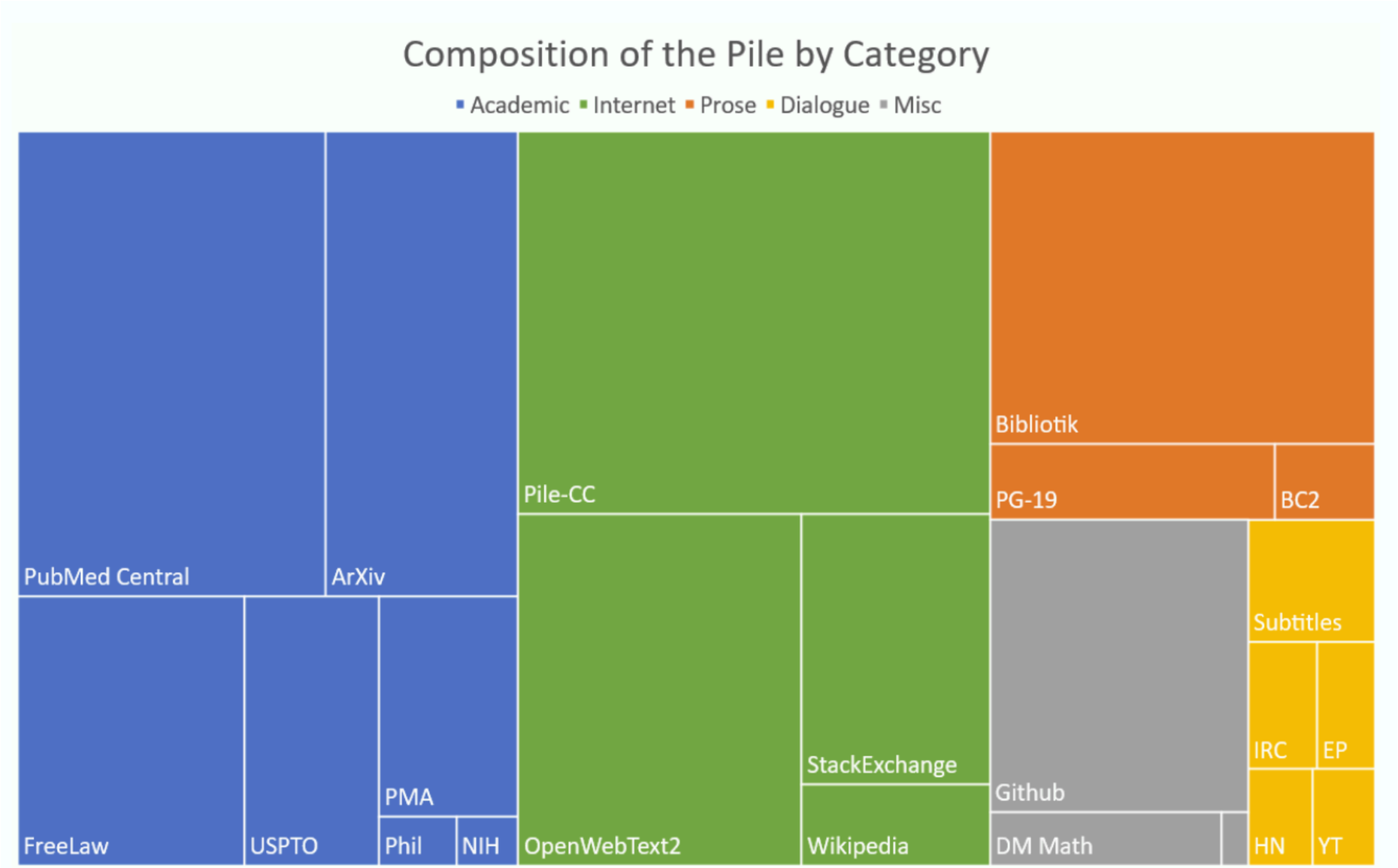


Data Preprocessing: Tokenization

- Most LLMs use byte-pair encoding
- Some experiments on simply using bytes as input, but this isn’t standard (why is this harder?)
- Speech language models
  - Vector Quantized codes (HuBERT)
  - Encodec or SSL-based discrete codes

Pretraining Text Datasets

If you want to train a language model from scratch, no need to scrape the Internet yourself!

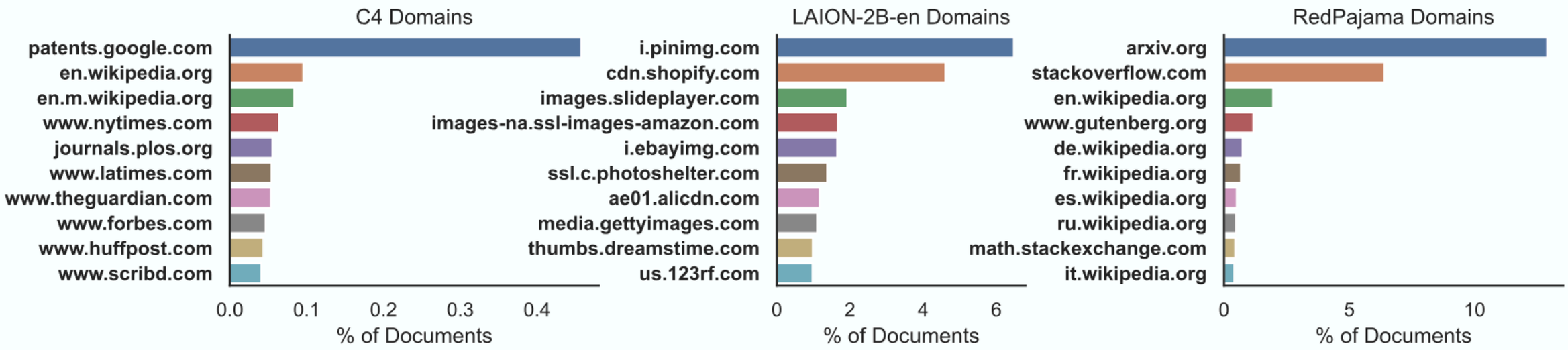


Gao et al. 2020

Pretraining Datasets

If you want to train a language model from scratch, no need to scrape the Internet yourself!

Dataset	Origin	Model	Size (GB)	# Documents	# Tokens
OpenWebText	Gokaslan & Cohen (2019)	GPT-2* (Radford et al., 2019)	41.2	8,005,939	7,767,705,349
C4	Raffel et al. (2020)	T5 (Raffel et al., 2020)	838.7	364,868,892	153,607,833,664
mC4-en	Chung et al. (2023)	umT5 (Chung et al., 2023)	14,694.0	3,928,733,374	2,703,077,876,916
OSCAR	Abadji et al. (2022)	BLOOM* (Scao et al., 2022)	3,327.3	431,584,362	475,992,028,559
The Pile	Gao et al. (2020)	GPT-J/Neo & Pythia (Biderman et al., 2023)	1,369.0	210,607,728	285,794,281,816
RedPajama	Together Computer (2023)	LLaMA* (Touvron et al., 2023)	5,602.0	930,453,833	1,023,865,191,958
S2ORC	Lo et al. (2020)	SciBERT* (Beltagy et al., 2019)	692.7	11,241,499	59,863,121,791
peS2o	Soldaini & Lo (2023)	-	504.3	8,242,162	44,024,690,229
LAION-2B-en	Schuhmann et al. (2022)	Stable Diffusion* (Rombach et al., 2022)	570.2	2,319,907,827	29,643,340,153
The Stack	Kocetkov et al. (2023)	StarCoder* (Li et al., 2023)	7,830.8	544,750,672	1,525,618,728,620



Elazar et al. 2023

What are we training on?

- Filtering out copyrighted data is difficult
  - This data is also really high-quality, and there's a lot of it…
- Private companies have no incentive to release their training data

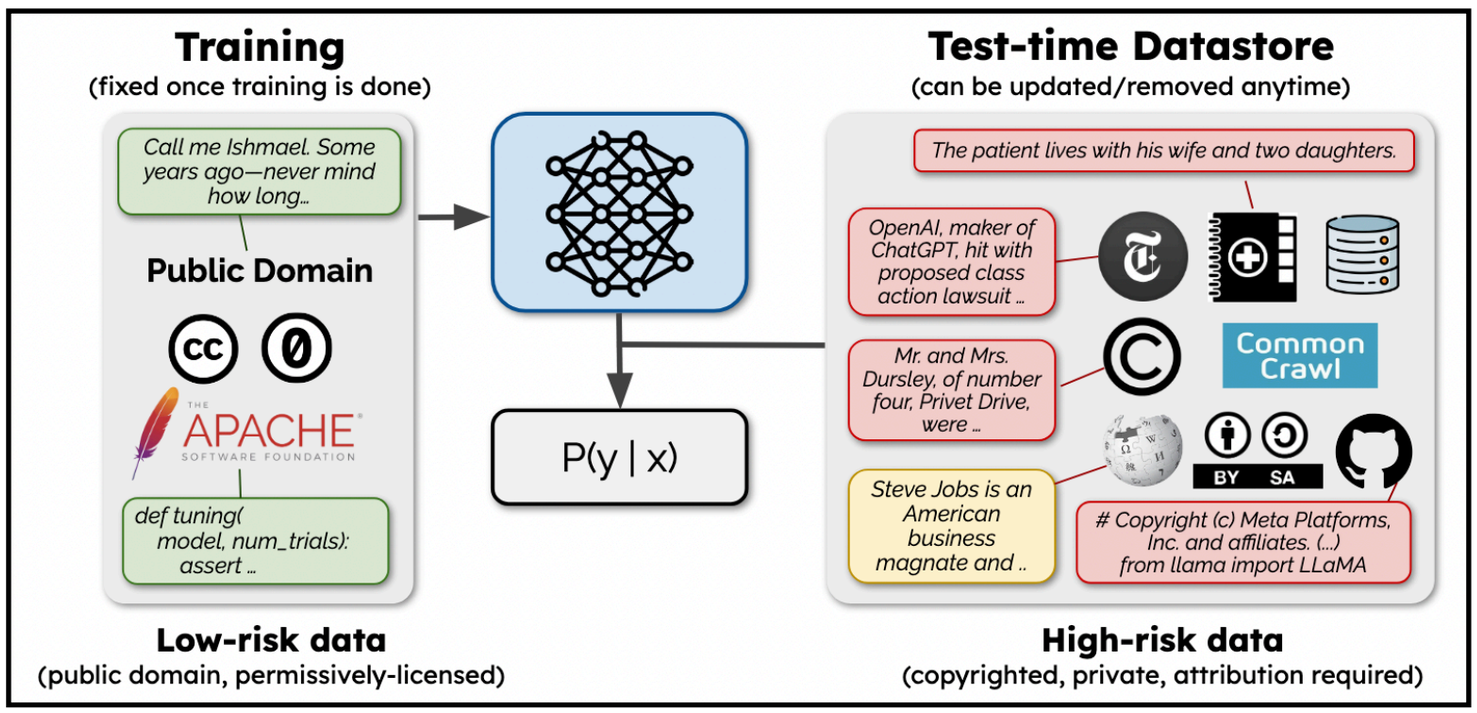
> Plaintiffs and defendants in copyright lawsuits over generative AI often make sweeping, opposing claims about the extent to which large language models (LLMs) have memorized plaintiffs' protected expression in their training data. Drawing on both machine learning and copyright law, we show that these polarized positions dramatically oversimplify the relationship between memorization and copyright. To do so, we extend a recent probabilistic extraction technique to measure memorization of 50 books in 17 open-weight LLMs. Through thousands of experiments, we show that the extent of memorization varies both by model and by book. With respect to our specific extraction methodology, we find that most LLMs do not memorize most books—either in whole or in part. However, we also find that LLAMA 3.1 70B entirely memorizes some books, like the first *Harry Potter* book and *1984*. In fact, the first *Harry Potter* is so memorized that, using a seed prompt consisting of just the first few tokens of the first chapter, we can deterministically generate the entire book near-verbatim. We discuss why our results have significant implications for copyright cases, though not ones that unambiguously favor either side.

Pamela Samuelson, prof in Berkeley Law — talk next week (Nov 10)

Cooper et al. 2025

What are we training on?

- Main premise: pretrain only on data we know is fair use
- Also pretrain the model to learn how to use a datastore of temporarily available documents
- At inference time, data the datastore can be updated to include / exclude sensitive information, copyrighted documents, etc.



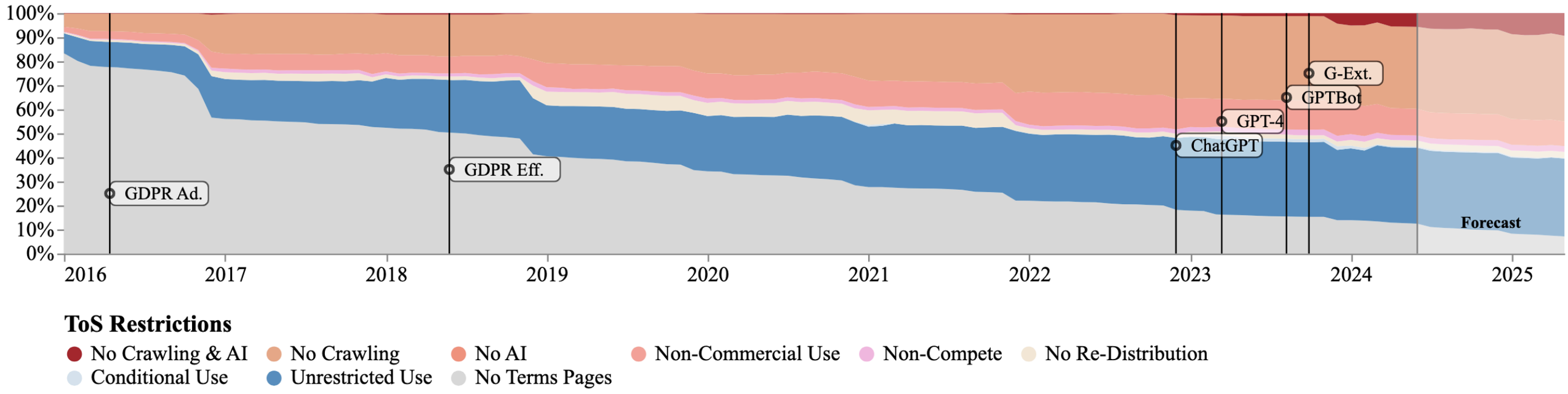
SILO Language Model

Min et al. 2023

What aren’t we training on?

- Low-resource languages or dialects (e.g., African American English)
- Non-written languages (e.g., American Sign Language)
- Language from people who aren’t on the web (e.g., older adults)
- Data that is increasingly getting excluded from scraping, for better or worse

This leads to language technologies that only work for some people!

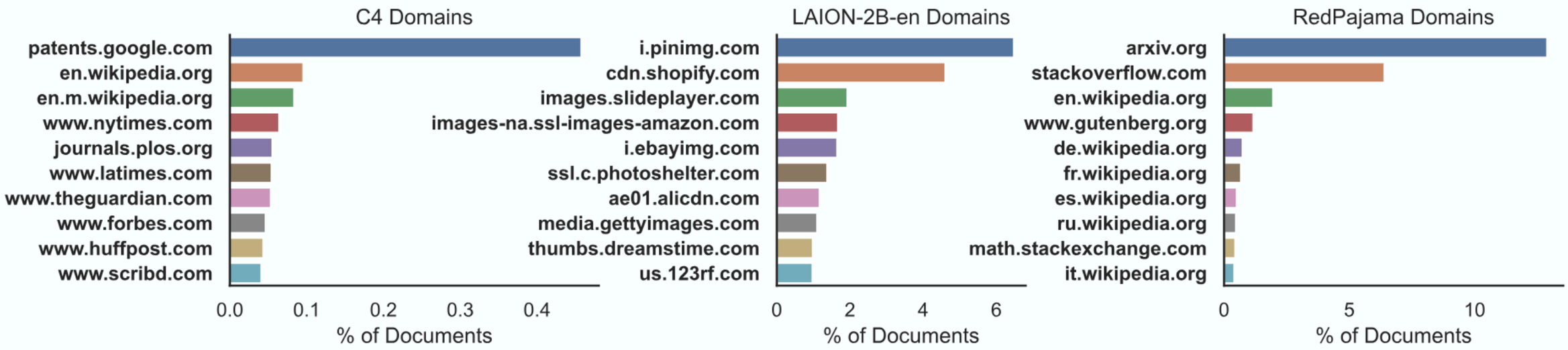


Longpre et al. 2024

Pretraining Datasets

If you want to train a language model from scratch, no need to scrape the Internet yourself!

Dataset	Origin	Model	Size (GB)	# Documents	# Tokens
OpenWebText	Gokaslan & Cohen (2019)	GPT-2* (Radford et al., 2019)	41.2	8,005,939	7,767,705,349
C4	Raffel et al. (2020)	T5 (Raffel et al., 2020)	838.7	364,868,892	153,607,833,664
mC4-en	Chung et al. (2023)	umT5 (Chung et al., 2023)	14,694.0	3,928,733,374	2,703,077,876,916
OSCAR	Abadji et al. (2022)	BLOOM* (Scao et al., 2022)	3,327.3	431,584,362	475,992,028,559
The Pile	Gao et al. (2020)	GPT-J/Neo & Pythia (Biderman et al., 2023)	1,369.0	210,607,728	285,794,281,816
RedPajama	Together Computer (2023)	LLaMA* (Touvron et al., 2023)	5,602.0	930,453,833	1,023,865,191,958
S2ORC	Lo et al. (2020)	SciBERT* (Beltagy et al., 2019)	692.7	11,241,499	59,863,121,791
peS2o	Soldaini & Lo (2023)	-	504.3	8,242,162	44,024,690,229
LAION-2B-en	Schuhmann et al. (2022)	Stable Diffusion* (Rombach et al., 2022)	570.2	2,319,907,827	29,643,340,153
The Stack	Kocetkov et al. (2023)	StarCoder* (Li et al., 2023)	7,830.8	544,750,672	1,525,618,728,620



Elazar et al. 2023

Pretraining Speech Models

Librispeech (960h)/
Librilite (60Kh)

English Pretraining

Model	Data	Objective	Input	Architectur	Params
wav2vec	960 / 60K	Masked	Waveform	Transformer	95M / 316M
HuBERT	960 / 60K	Pseudo-phone	Waveform	Transformer	95M / 316M
w2v-	60K	Masked	Spectrogra	Conformer	600M / 1B
WavLM	960 / 94K	Denoising +	Waveform	Transformer	95M / 316M
BEST-RQ	60K	RQ MLM	Spectrogra	Conformer	600M
Data2vec	960 / 60K	Masked Self-	Waveform	Transformer	95M / 316M

Emilia (101K Hrs) Multilingual Pretraining

Model	Param	Hours	Lang	Architecture	Objective
XLSR 53	316M	56K	53	Transformer	wav2vec 2.0
XLS-R 128	316M /	436K	128	Transformer	wav2vec 2.0
MMS	316M /	491K	1406	Transformer	wav2vec 2.0
Google	2B	12.5M	300	Conformer	BEST-RQ
w2v-BERT	600M	1M	143	Conformer	w2v-BERT
w2v-BERT	600M	4.5M	143	Conformer	W2v-BERT
XEUS	600M	1M	4057	E-Branchformer	WavLM-like

From Shinji Watanabe

What are we training on?

- Filtering out copyrighted data is difficult
  - This data is also really high-quality, and there's a lot of it…
- Private companies have no incentive to release their training data

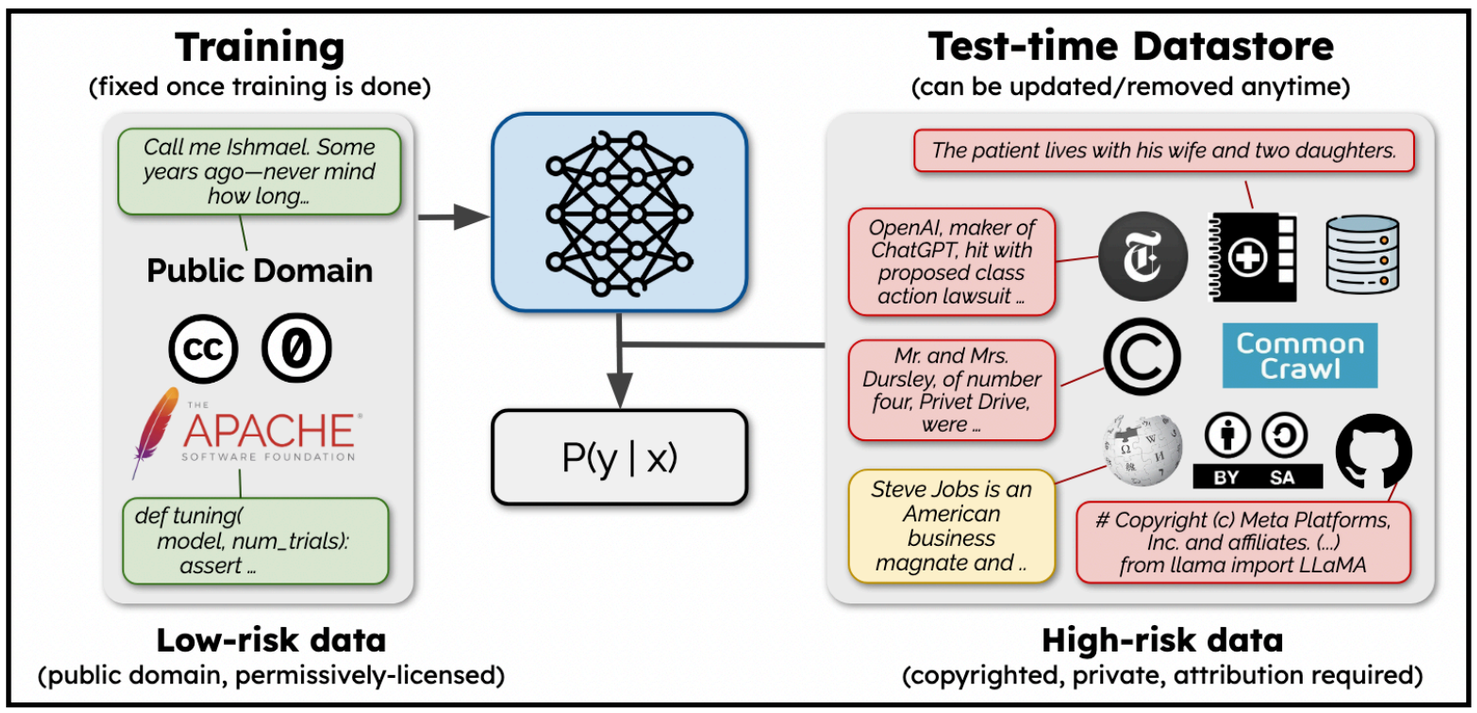
> Plaintiffs and defendants in copyright lawsuits over generative AI often make sweeping, opposing claims about the extent to which large language models (LLMs) have memorized plaintiffs' protected expression in their training data. Drawing on both machine learning and copyright law, we show that these polarized positions dramatically oversimplify the relationship between memorization and copyright. To do so, we extend a recent probabilistic extraction technique to measure memorization of 50 books in 17 open-weight LLMs. Through thousands of experiments, we show that the extent of memorization varies both by model and by book. With respect to our specific extraction methodology, we find that most LLMs do not memorize most books—either in whole or in part. However, we also find that LLAMA 3.1 70B entirely memorizes some books, like the first *Harry Potter* book and *1984*. In fact, the first *Harry Potter* is so memorized that, using a seed prompt consisting of just the first few tokens of the first chapter, we can deterministically generate the entire book near-verbatim. We discuss why our results have significant implications for copyright cases, though not ones that unambiguously favor either side.

Pamela Samuelson, prof in Berkeley Law — talk next week (Nov 10)

Cooper et al. 2025

What are we training on?

- Main premise: pretrain only on data we know is fair use
- Also pretrain the model to learn how to use a datastore of temporarily available documents
- At inference time, data the datastore can be updated to include / exclude sensitive information, copyrighted documents, etc.



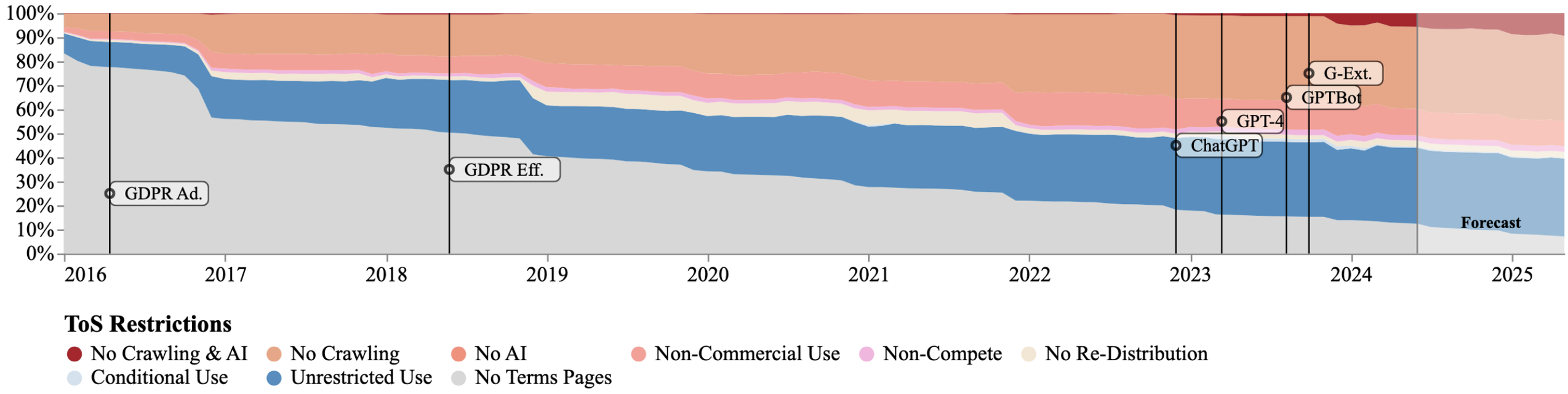
SILO Language Model

Min et al. 2023

What aren’t we training on?

- Low-resource languages or dialects (e.g., African American English)
- Non-written languages (e.g., American Sign Language)
- Language from people who aren’t on the web (e.g., older adults)
- Data that is increasingly getting excluded from scraping, for better or worse

This leads to language technologies that only work for some people!

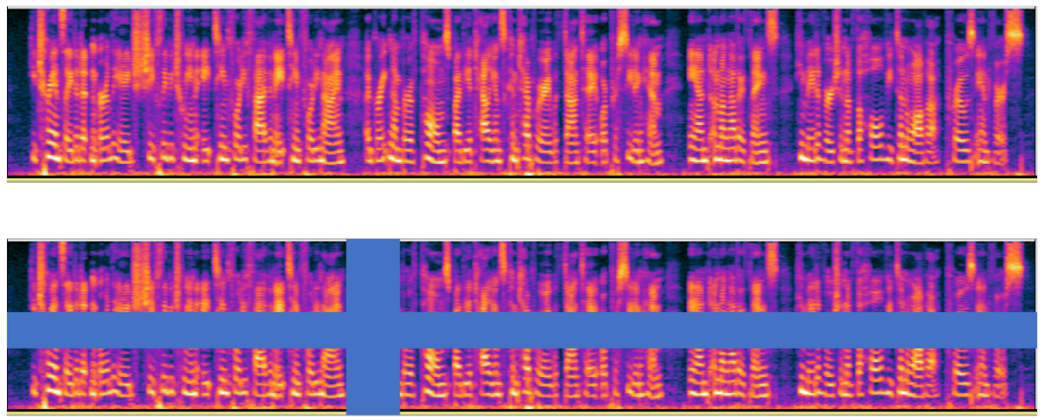


Longpre et al. 2024

In Speech we can augment training data

- In Speech Recognition some success was also obtained using SpecAugment [2019](Randomly masking parts of data)
- Although, at scale, these improvements are not as significant

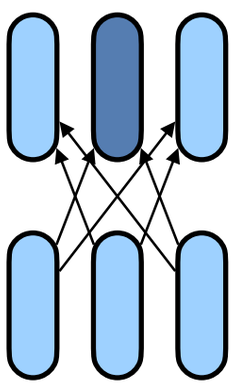
Specaugment (Masking)



ASR improvement

w/o specaug:	**9.8% WER**
w/ specaug:	5.8% WER

Pretraining Objectives



Encoder-only

Masked language modeling objective: probability of each word depends on every other word in the sequence

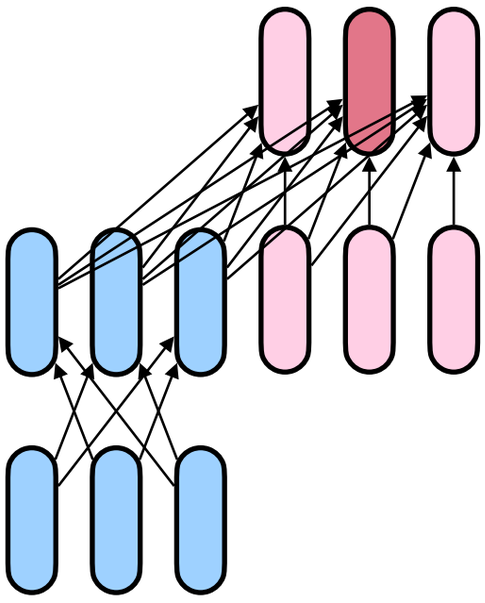
$$p(x_i) \propto f\left(\left\langle x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{|\overline{x}|} \right\rangle\right)$$

- Really useful for learning high-quality text embeddings, because the representations of each word depend on previous **and** future words

- Not useful for generating sequences token-by-token (why?)

- Representative model: **BERT** (Devlin et al. 2018)

Pretraining Objectives



Encoder-decoder
Each output token is dependent on all of the input tokens, and whatever outputs have been generated so far

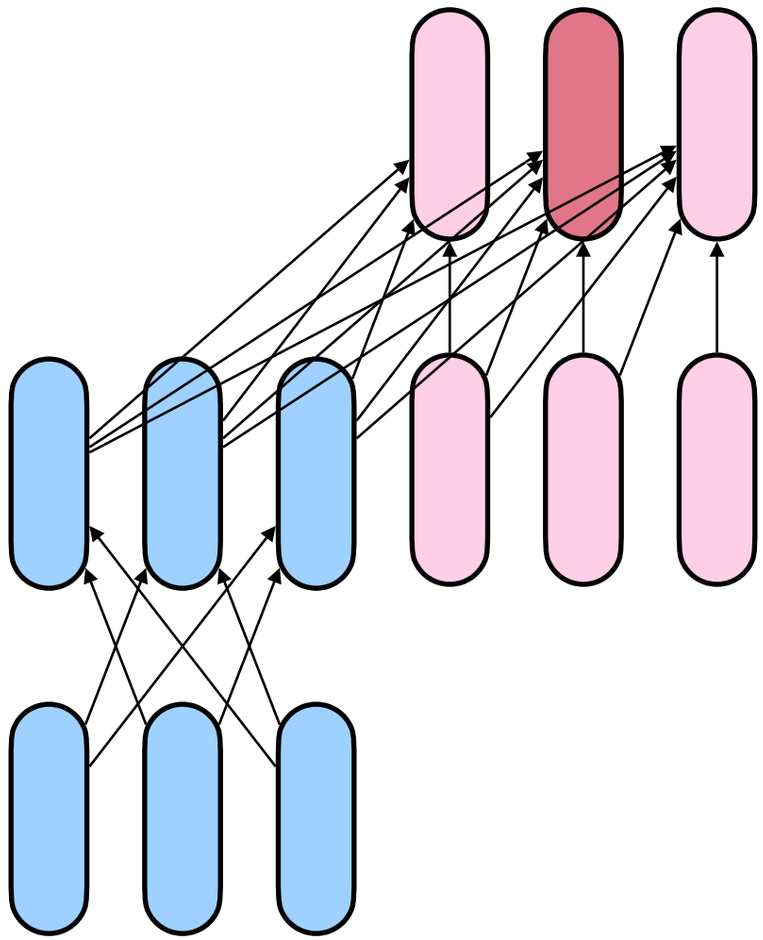
$$p(y_i) \propto f\left(\left\langle x_1, \ldots, x_{|\overline{x}|} \right\rangle, \left\langle y_1, \ldots, y_{i-1} \right\rangle\right)$$

- Input sequence is bidirectionally encoded, but a decoder can generate outputs token-by-token
- Training objective: span corruption and reconstruction

thank you ~~for inviting~~ me to your party ~~last~~ week
s1 s2

- Representative model: **T5** (Raffel et al. 2018)

Pretraining Objectives



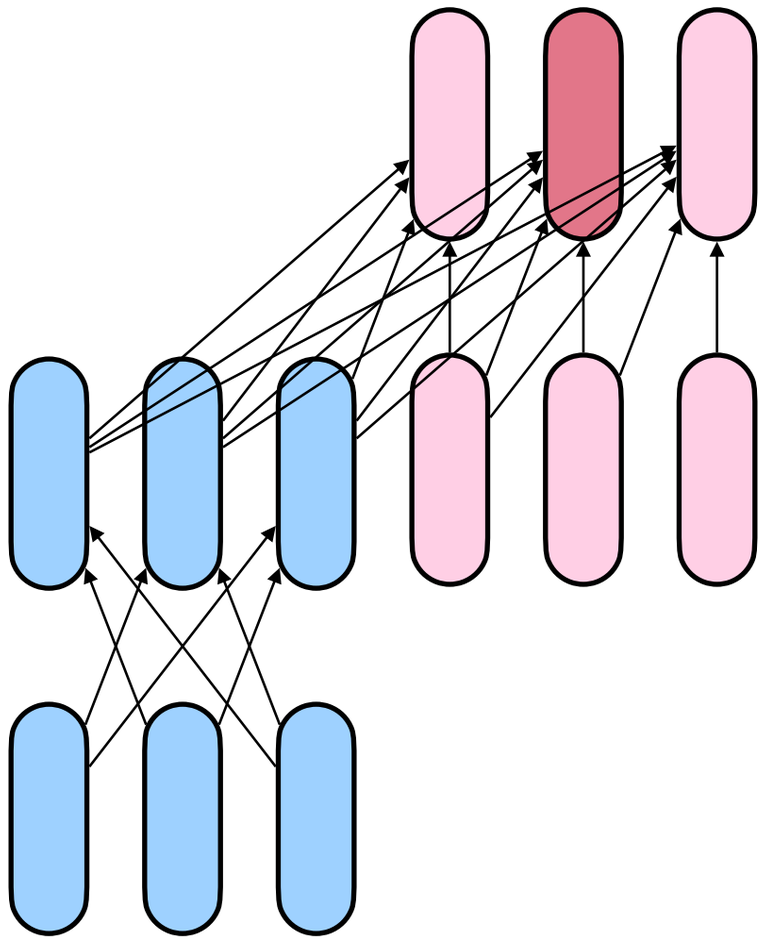
thank you ~~for inviting~~ me to your party ~~last~~ week

s1 s2

Pretraining Objectives

<s1> for inviting <s2> last

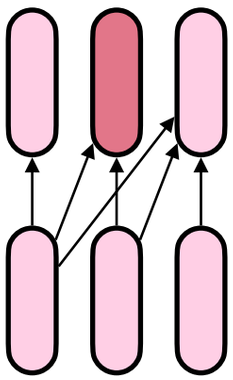
training target (autoregressive)



thank you <s1> *me to your party*<s2>*week*

input (fully encoded)

Pretraining Objectives

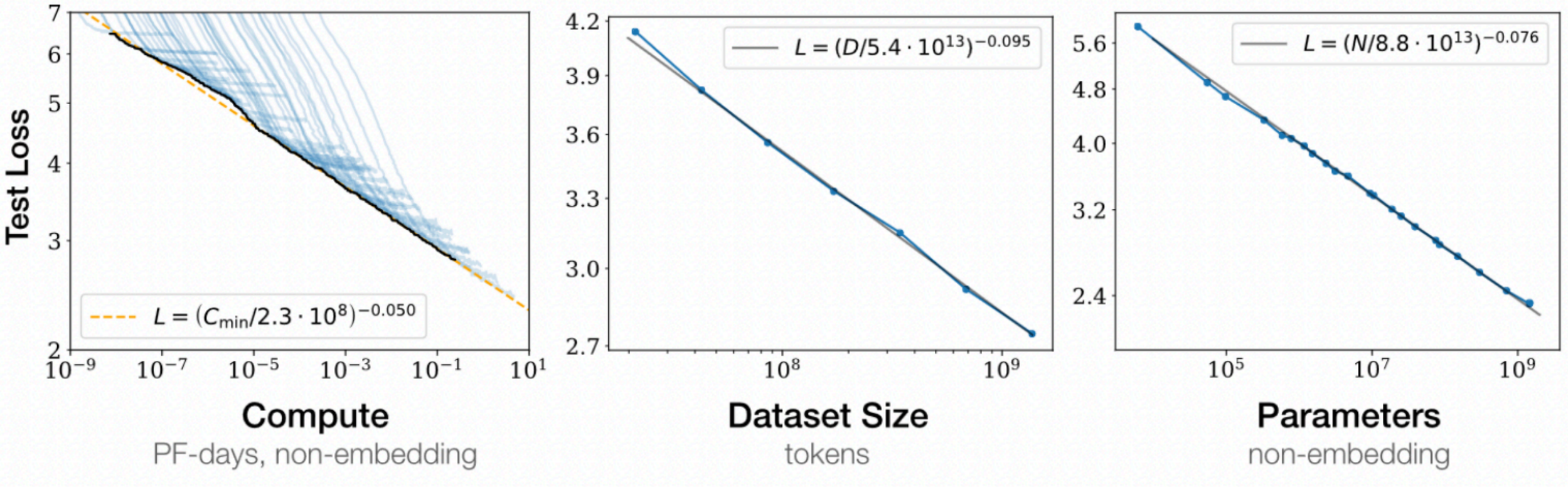


Decoder-only
Probability of each word depends only on the previous tokens generated so far

$$p(y_i) \propto f\left(\langle y_1, \ldots, y_{i-1} \rangle\right)$$

- Easy and fast token-by-token generation
- Basically all major pretrained language models are decoder-only models
- Representative model: **GPT series** (Radford et al. 2018)

Scaling Up



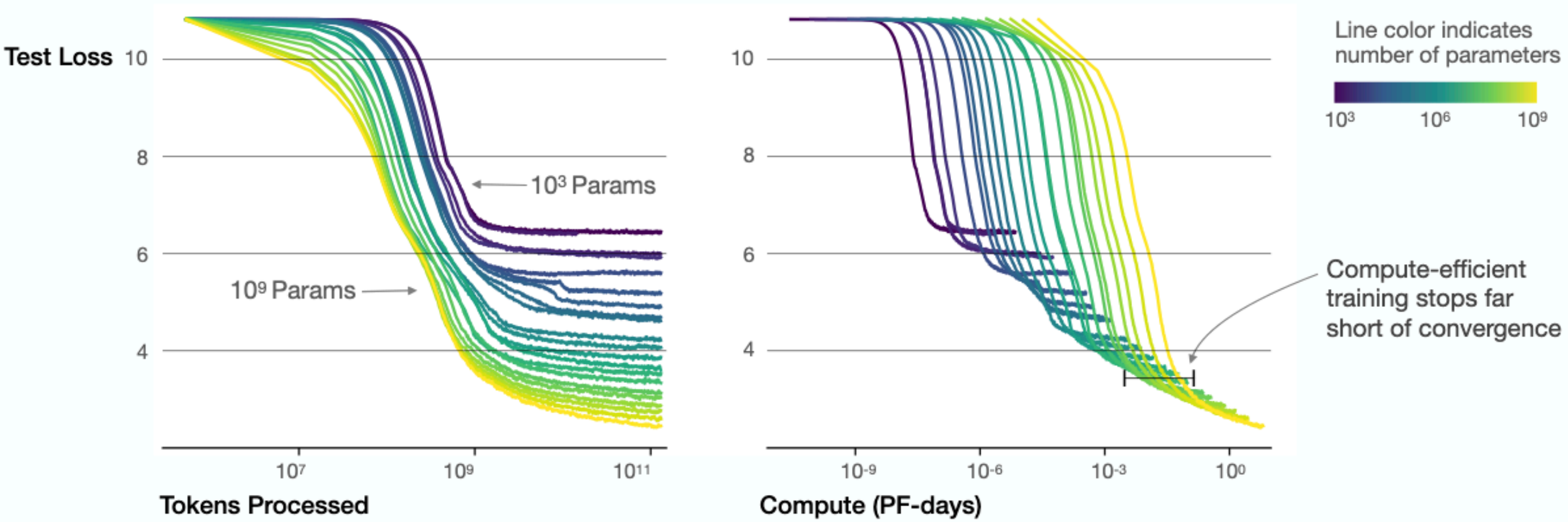
Better at language modeling when we train for longer, increase our dataset size, and increase the model size

Scaling laws tell us what test loss to expect given the amount of compute, data, and parameters.

- Discussion question: why non-embedding parameters?

Kaplan et al. 2020

Scaling Up

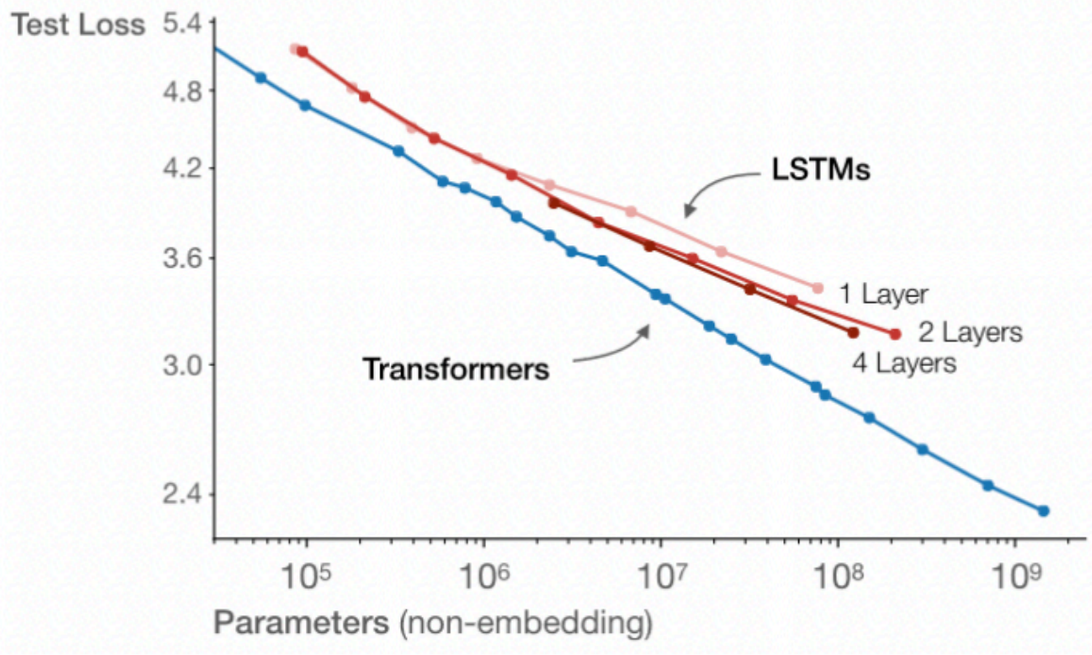


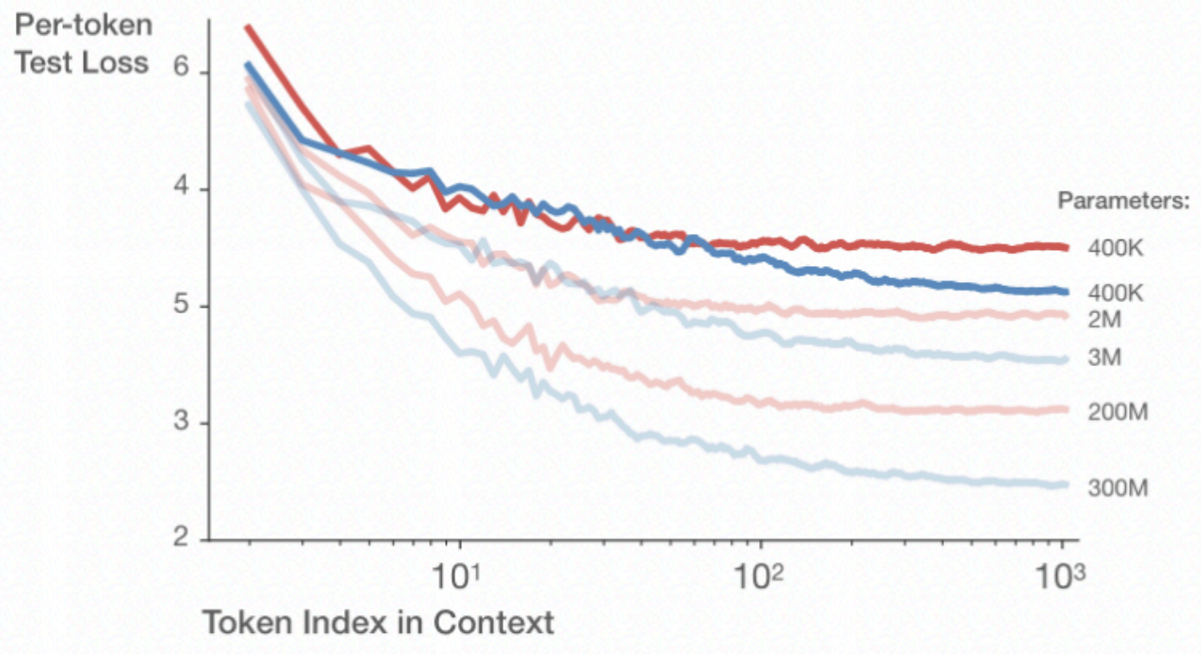
Better at language modeling when we train for longer, increase our dataset size, and increase the model size

Large models requires fewer samples to reach the same performance

Kaplan et al. 2020

Scaling Up



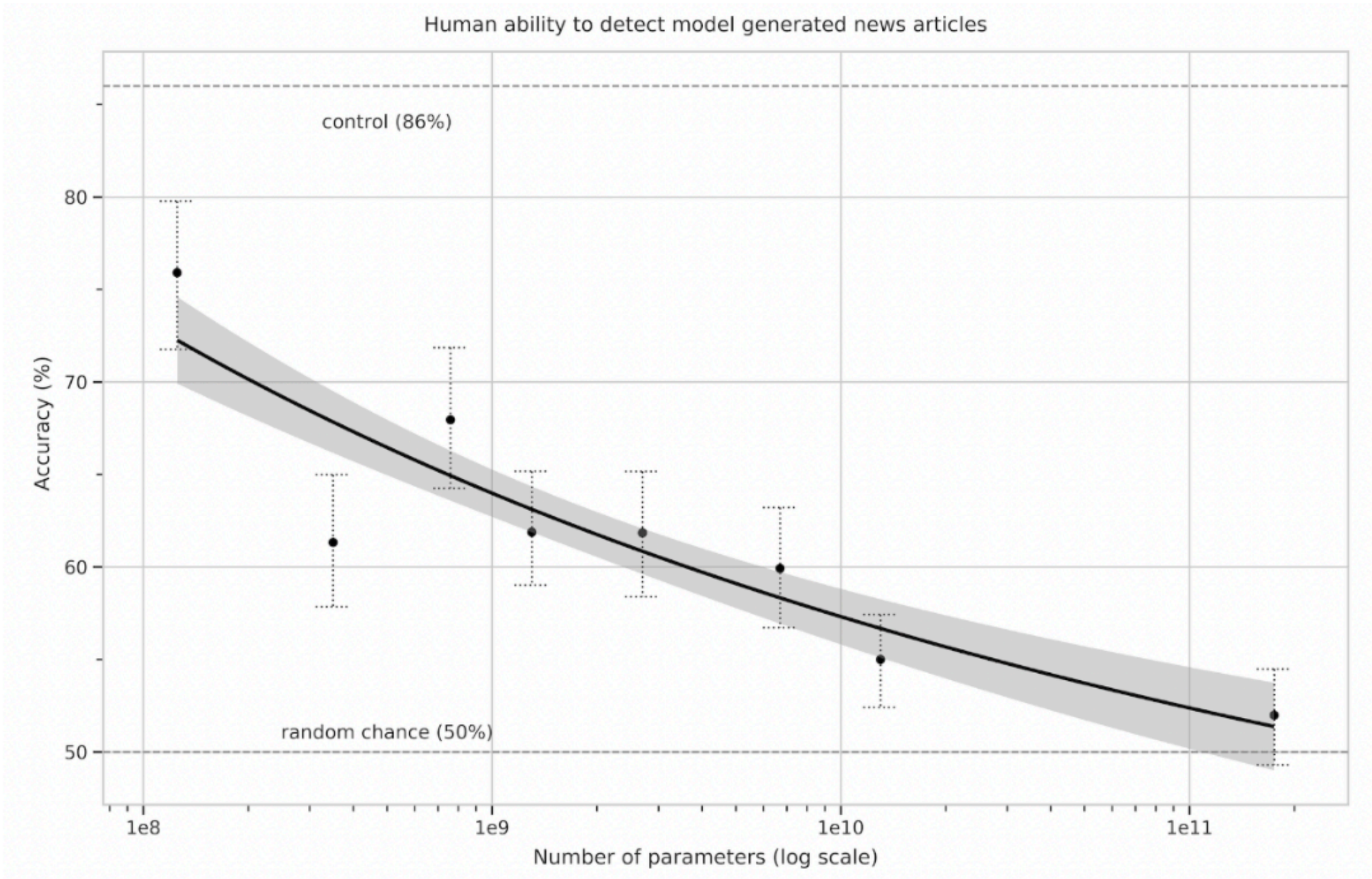


Better at language modeling when we train for longer, increase our dataset size, and increase the model size

LSTMs are limited by their ability to model long contexts

Kaplan et al. 2020

Scaling Up



Brown et al. 2020

Scaling Up

- One conclusion: we can reliably improve performance if we keep scaling up (data, model size, time for training)
  - (Where) might we hit a limit?
- Another conclusion: we can experiment with smaller models, and trends will probably generalize to larger models
  - Useful for prototyping, before spending millions of dollars on a pretraining run

Training Costs	Pre-Training	Context Extension	Post-Training	Total
in H800 GPU Hours	2664K	119K	5K	2788K
in USD	$5.328M	$0.238M	$0.01M	$5.576M

Table 1 | Training costs of DeepSeek-V3, assuming the rental price of H800 is $2 per GPU hour.

Menu of Pretrained Language Models

Zuckerberg says Meta will build data center the size of Manhattan in latest AI push

CEO says company plans to spend hundreds of billions on developing artificial intelligence products

Llama series: popular open-weight language model from Meta

			Parameters	Training cost (petaFLOP-day)	Context length (tokens)	Corpus size (tokens)
		tinued	• 6.7B • 13B • 32.5B • 65.2B	6,300[44]	2048	1–1.4T
		tinued	• 6.7B • 13B • 69B	21,000[45]	4096	2T
Code Llama	August 24, 2023	Discontinued	• 6.7B • 13B • 33.7B • 69B	?		
Llama 3	April 18, 2024	Active	• 8B • 70.6B	100,000[46][47]	8192	15T
Llama 3.1	July 23, 2024	Active	• 8B • 70.6B • 405B	440,000[37][48]	128,000	
Llama 3.2	September 25, 2024	Active	• 1B • 3B • 11B • 90B[49][50]	?	128,000[51]	9T
Llama 3.3	December 7, 2024	Active	• 70B	?	128,000	15T+
Llama 4	April 5, 2025	Active	• 109B • 400B • 2T	• 71,000 • 34,000 • ?[38]	• 10M • 1M • ?	• 40T • 22T • ?

https://github.com/meta-llama/llama-models, Wikipedia

Summary

- **Pretraining:** collect a bunch of web data, preprocess it, then apply the language modeling objective of your choice

- Some work hypothesizes **scaling laws** that govern the relationship between model size, amount of data, and training time, and the resulting language modeling ability

- **Keep in mind:** pretraining is really expensive, and the data you train on might not be yours to use!

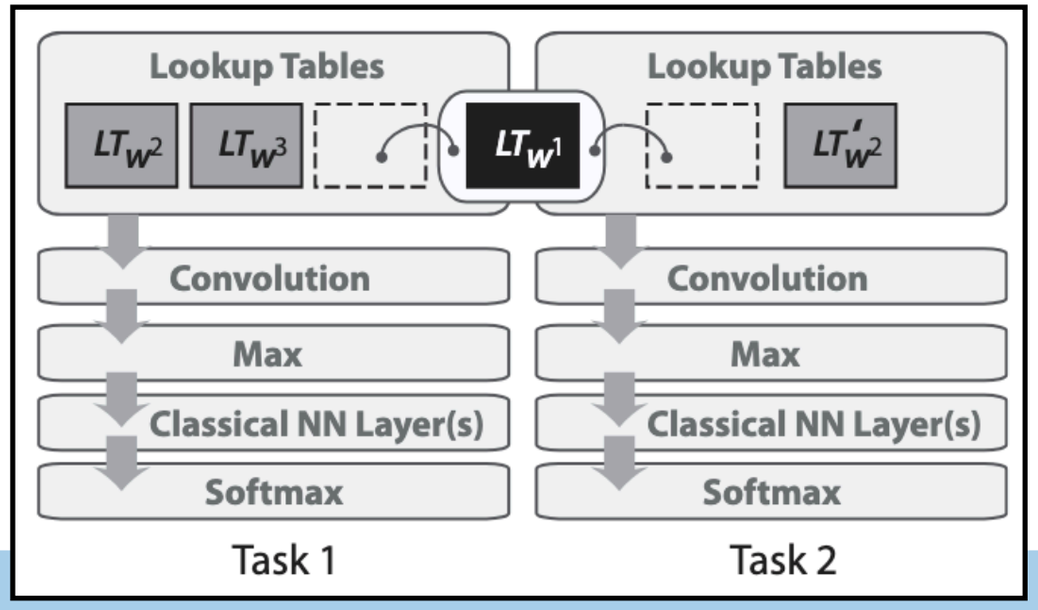
Single-Task NLP

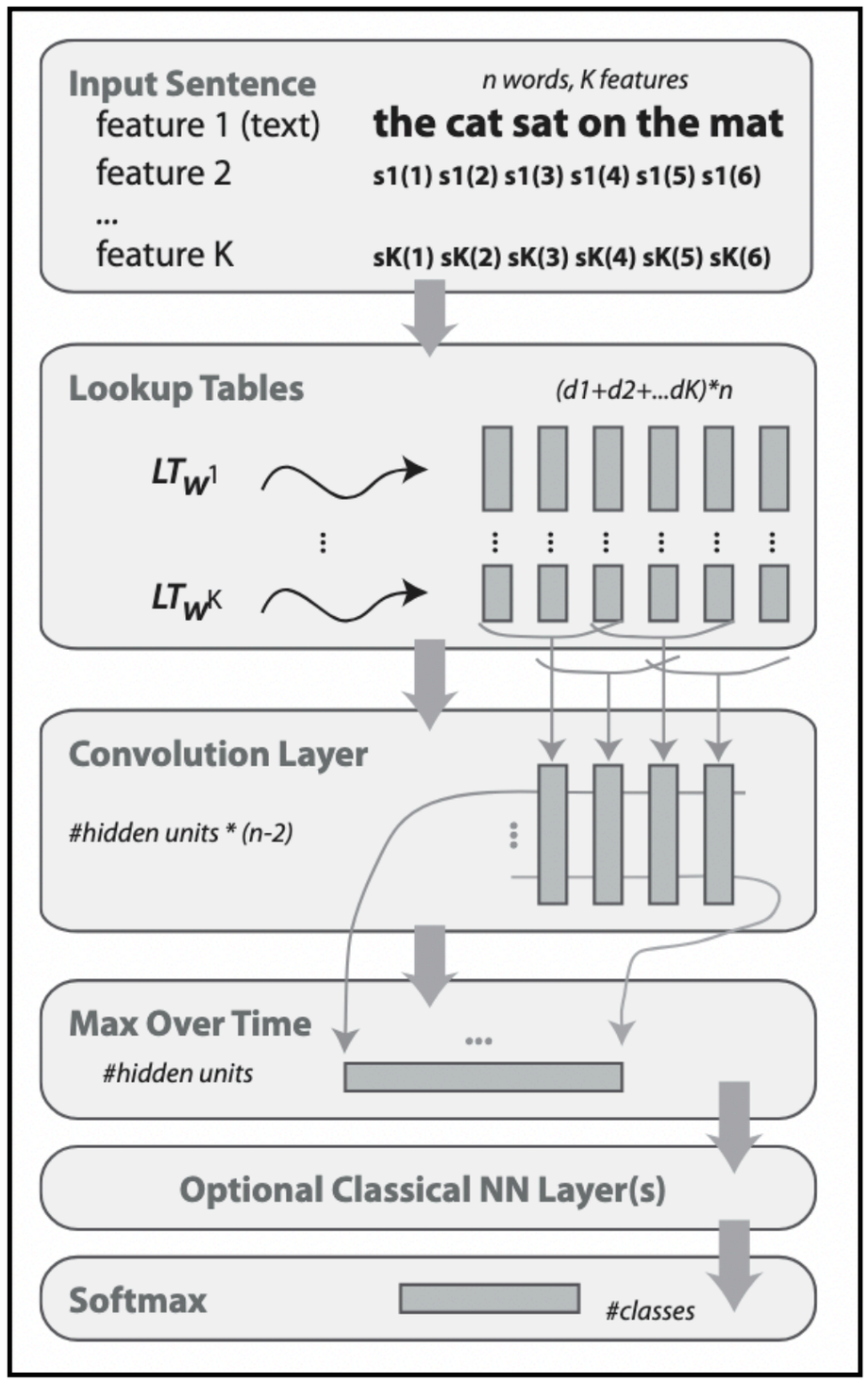
- NLP working assumptions pre-2018
  - We first need to understand the atomic units (which ones?), then we can study how they are composed to give rise to meaning
    - These compositional processes need to be modeled explicitly
    - If we want to do something beyond language modeling, we need to train a specialized model
    - Eventually, we can combine everything into an end-to-end dialogue system…
- There were some efforts to train natively multitask language models…

The Dream of Multitask NLP

- Map words to embeddings, including positional embeddings
- Convolution-based context processing for variable length sequences
- Multi-layer prediction for classification task
- End-to-end training via backpropagation on different NLP tasks (SRL, POS tagging, etc.)
- Leverage unlabeled data with a language modeling objective
- Why didn't we get GPT in 2008?

A General Deep Architecture for NLP (2008)





Collobert and Weston 2008

The Dream of Multitask NLP

knowledge → information extraction

Cal is in _____, California

I put _____ fork down on the table

syntax → POS tagging

The woman walked across the street, checking for traffic over _____ shoulder

coreference → coreference resolution

I went to the ocean to see the fish, turtles, seals, and _____

lexical semantics → word embeddings

Overall, the value I got from the two hours watching it was the sum total of the popcorn and the drink. The movie was ____

sentiment → sentiment classification

reasoning → question answering

I'm thinking about a sequence that goes 1, 2, 3, 5, 8, 13, 21, _____

Realizing the Dream

- **What happened to our pre-2018 assumptions?**

- What we learn from language modeling looks a lot like what need for traditional NLP tasks!

- Self-supervised approaches showed we might not need to independently learn word and sentence representations

- (In fact, we can recover a lot of the structural features we were explicitly modeling before from these representations!)

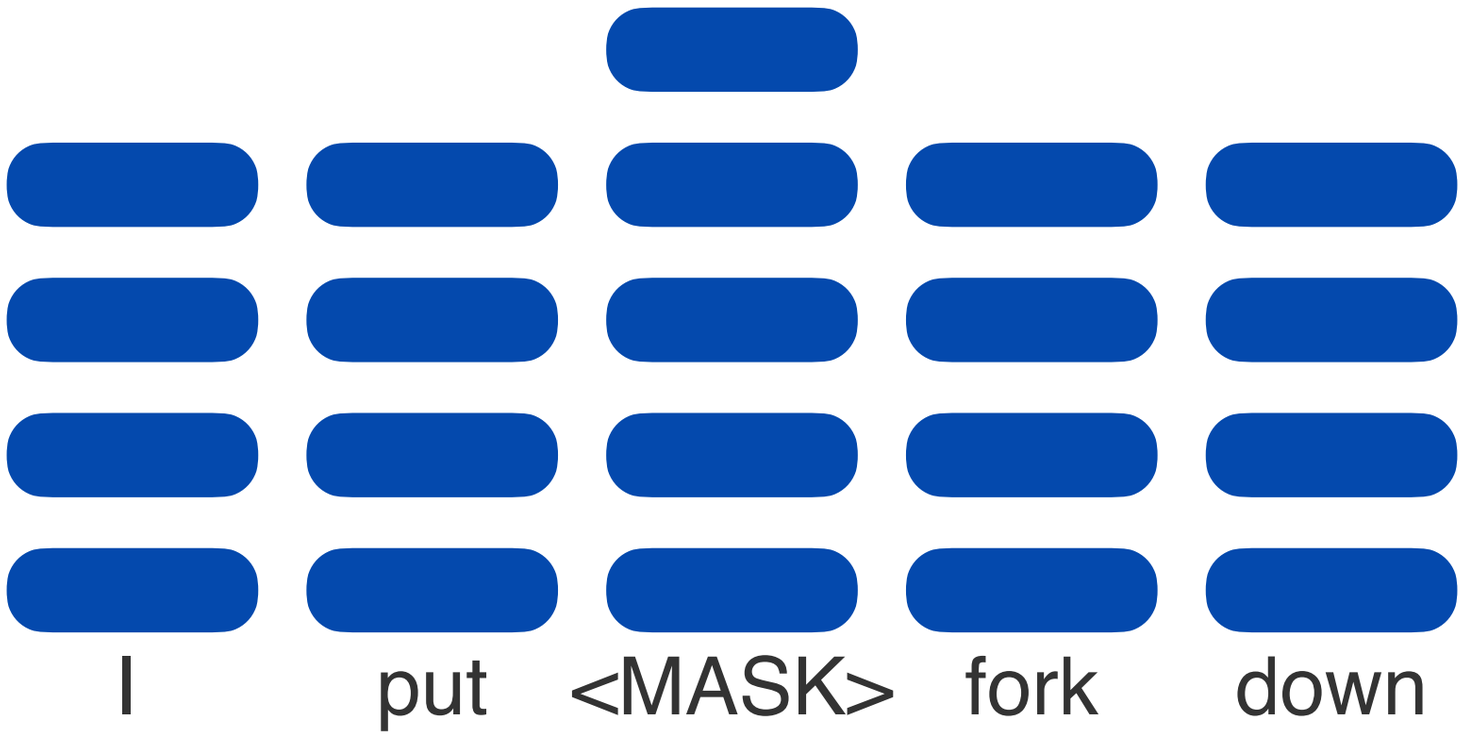
Probing

- Main research question:
  - We've trained an end-to-end language model that isn't a combination of a bunch of NLP tasks
    - But were these tasks still useful for the model to learn?
- How do we know what the model has "learned", besides what we test it on?
- Method: probing

Probing

- As we run inference on a model to predict the next word, we compute intermediate vector representations (e.g., values at different transformer layers)

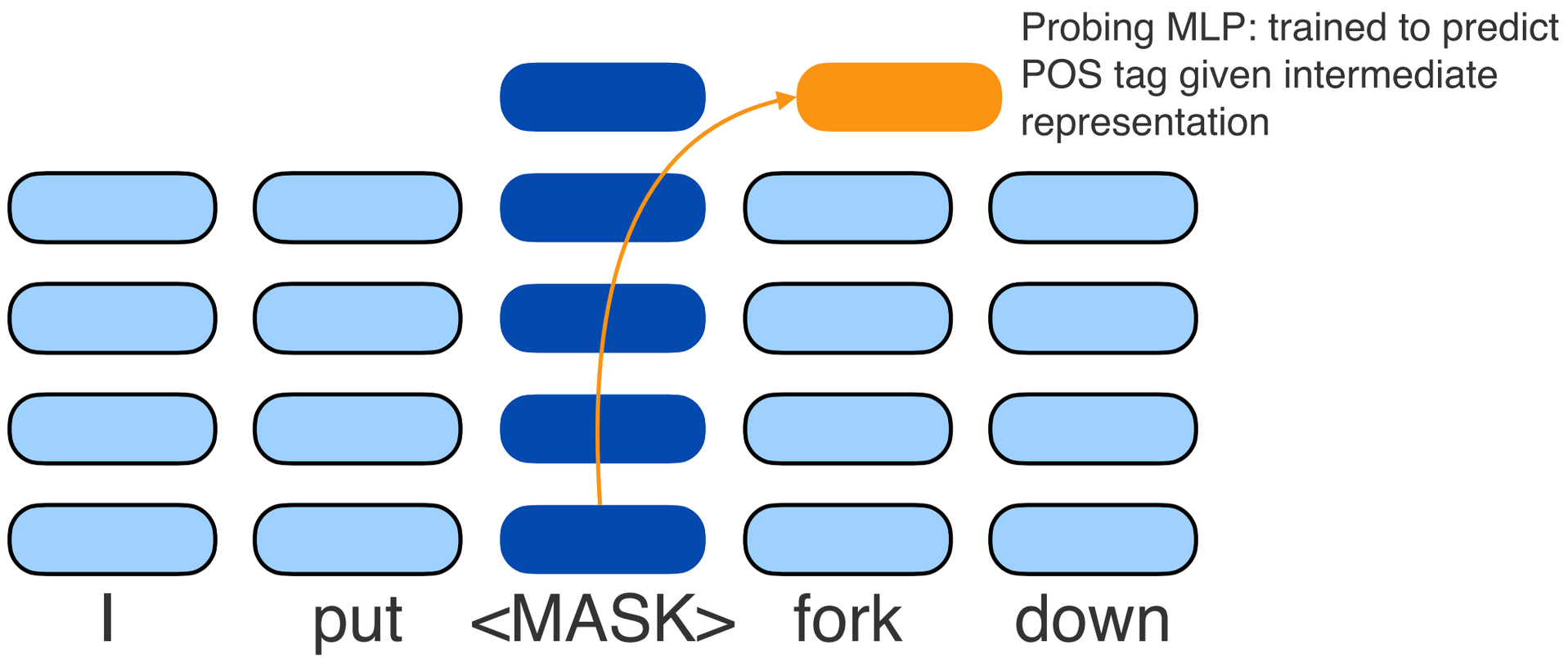
- Do these intermediate values contain the information relevant to an NLP task?



Probing

- As we run inference on a model to predict the next word, we compute intermediate vector representations (e.g., values at different transformer layers)

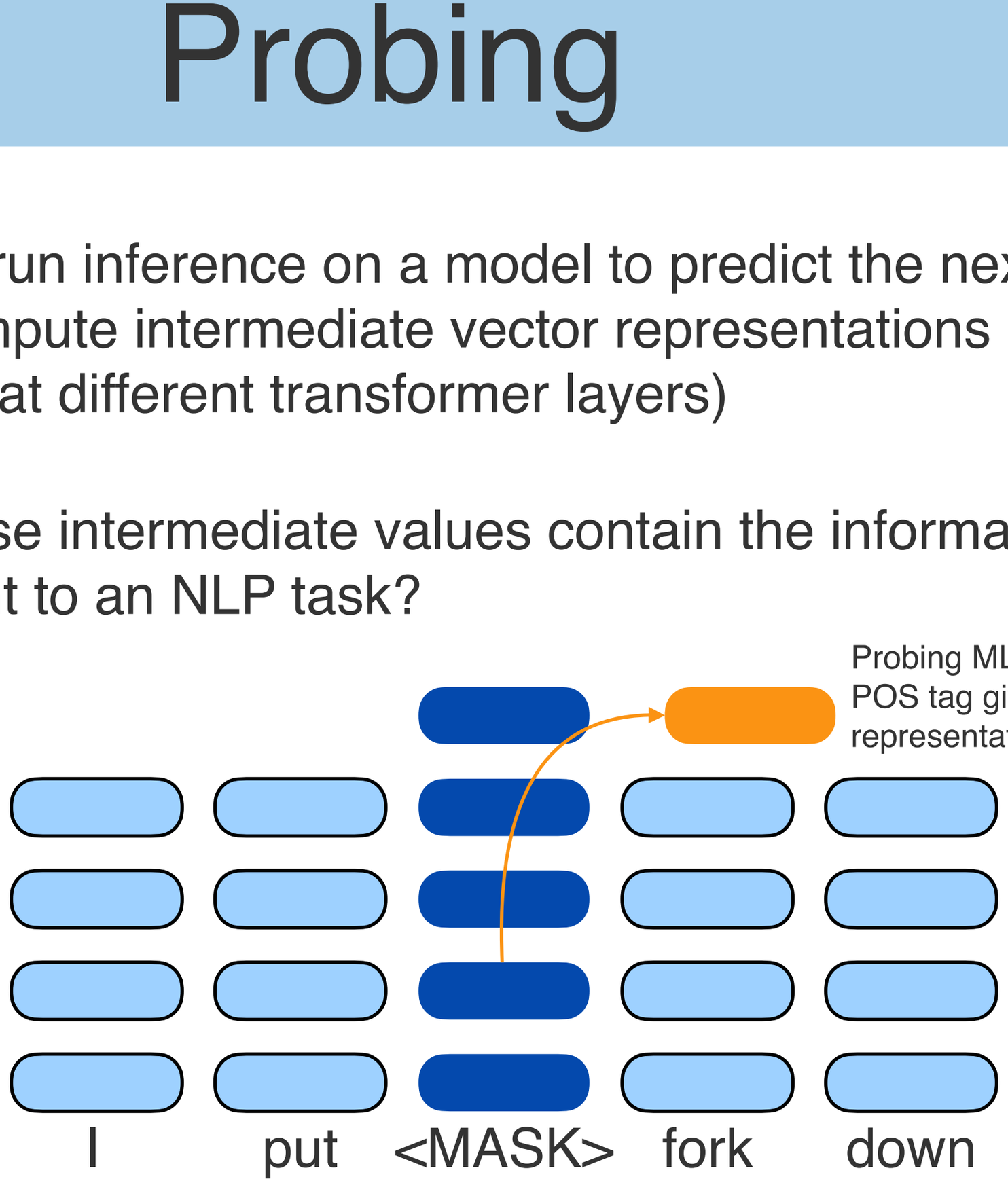
- Do these intermediate values contain the information relevant to an NLP task?



Probing

- As we run inference on a model to predict the next word, we compute intermediate vector representations (e.g., values at different transformer layers)

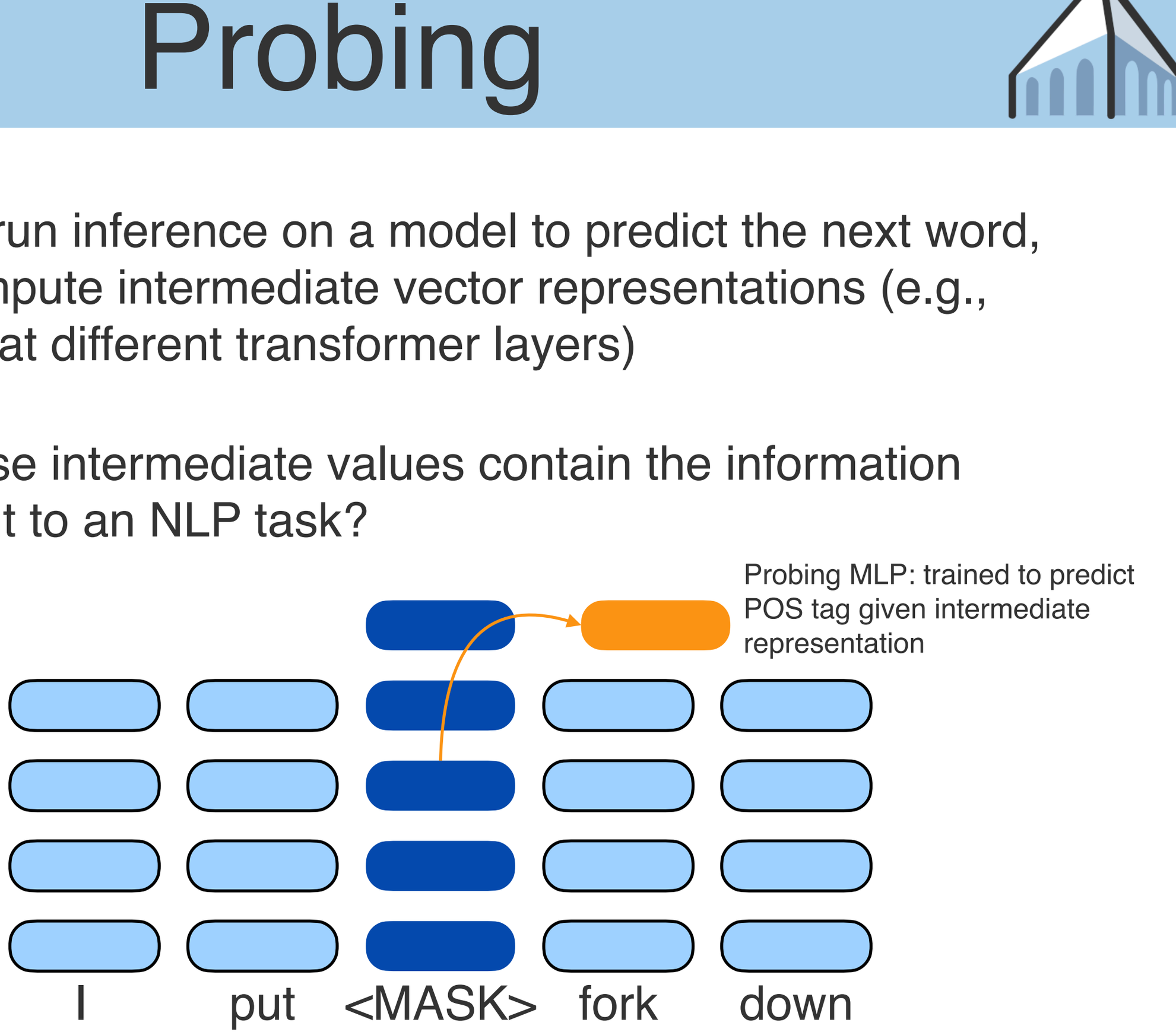
- Do these intermediate values contain the information relevant to an NLP task?



Probing

- As we run inference on a model to predict the next word, we compute intermediate vector representations (e.g., values at different transformer layers)

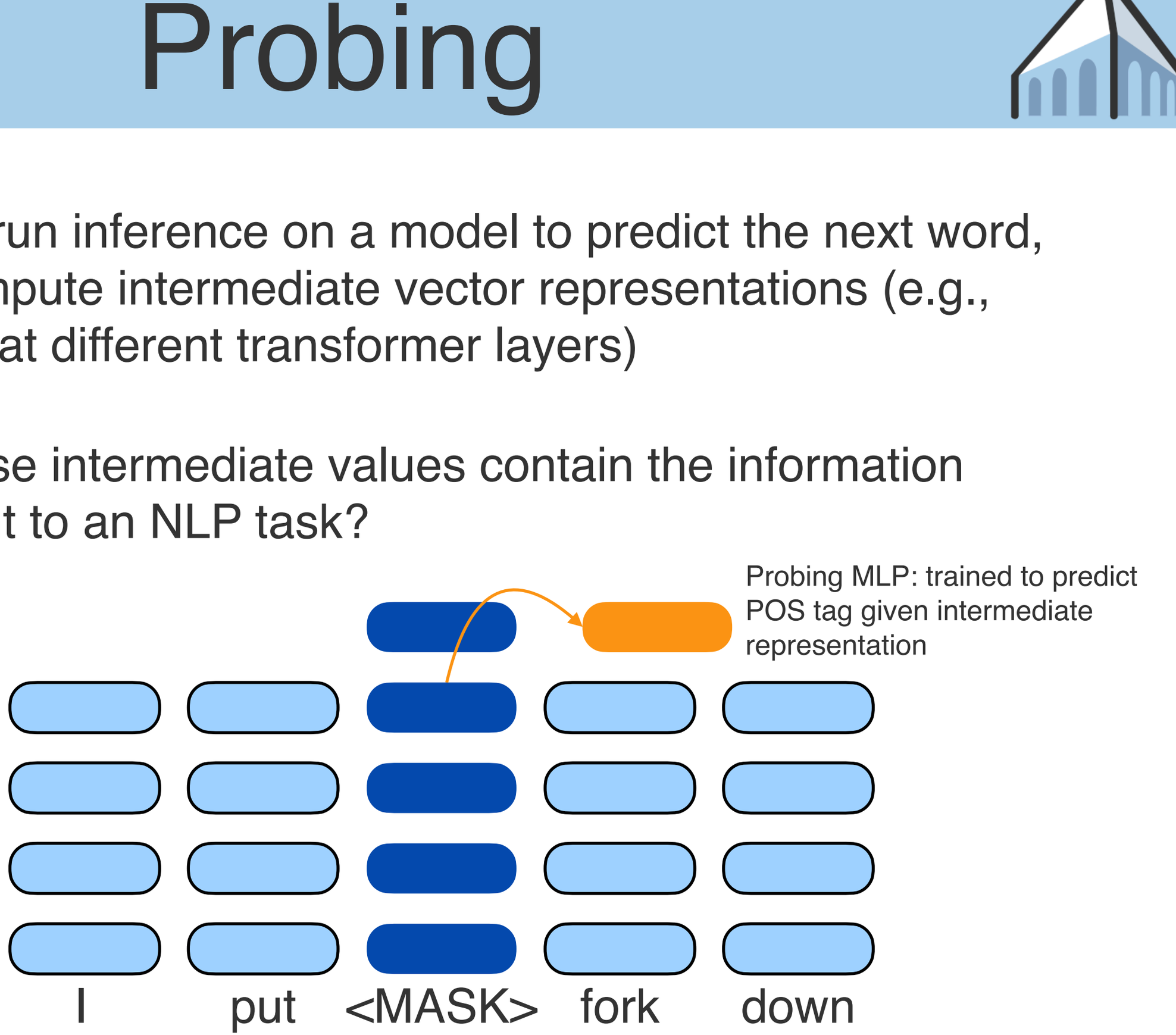
- Do these intermediate values contain the information relevant to an NLP task?



Probing

- As we run inference on a model to predict the next word, we compute intermediate vector representations (e.g., values at different transformer layers)

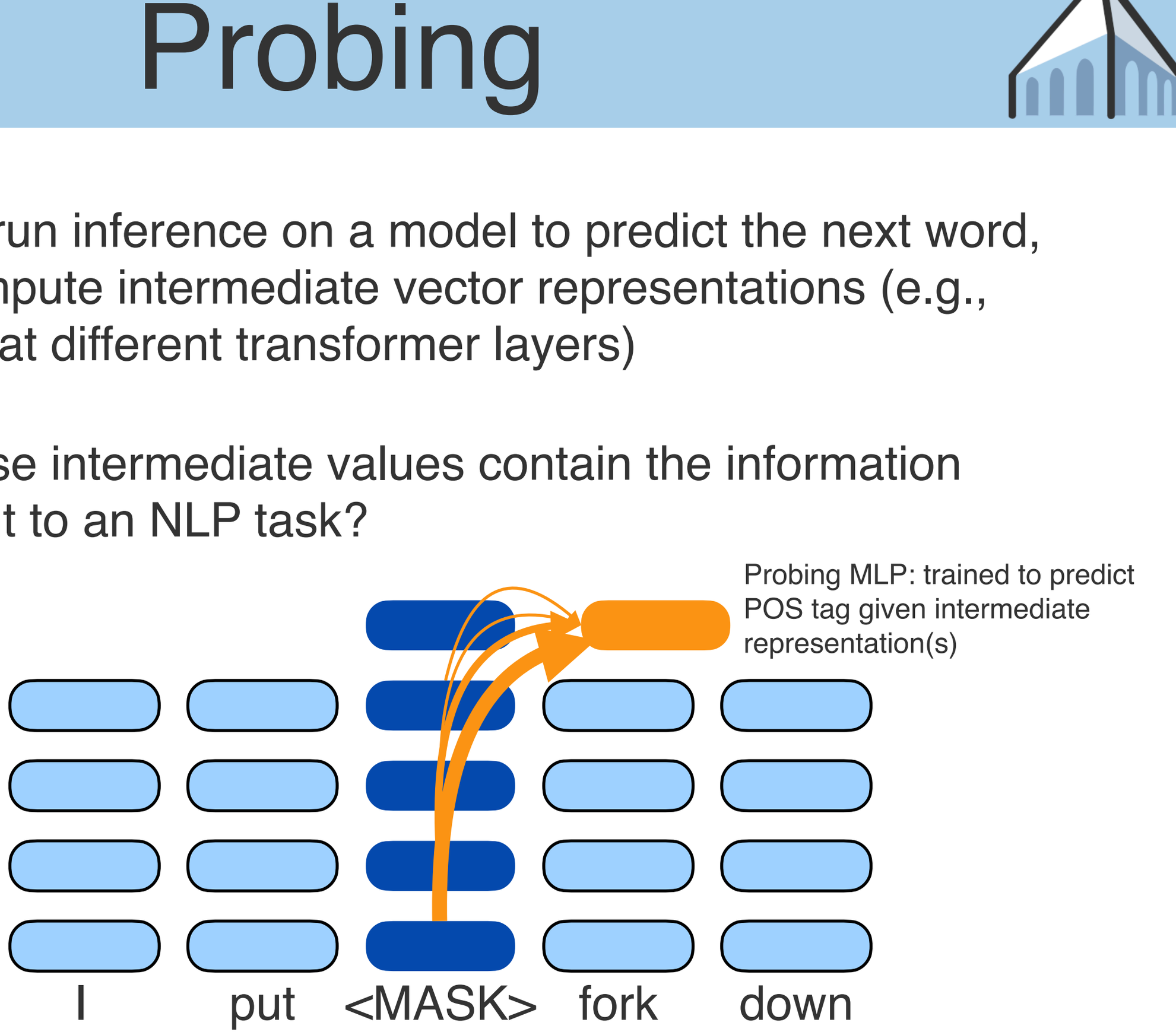
- Do these intermediate values contain the information relevant to an NLP task?



Probing

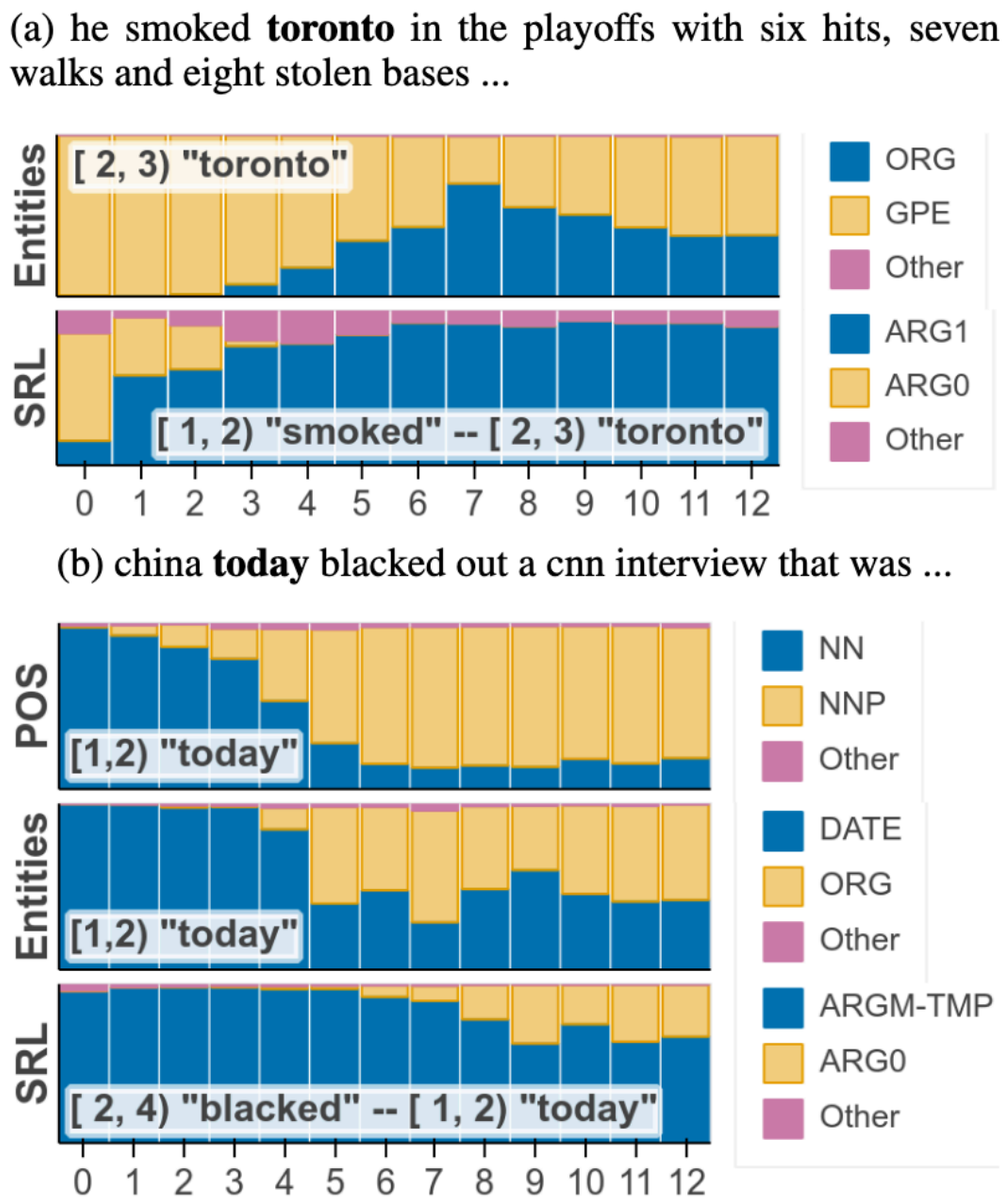
- As we run inference on a model to predict the next word, we compute intermediate vector representations (e.g., values at different transformer layers)

- Do these intermediate values contain the information relevant to an NLP task?



Probing

- "BERT rediscovers the classical NLP pipeline" (2019)
- Intermediate representations of BERT at different layers contain sufficient information to perform well on NLP tasks, without any task-specific training!



Tenney et al. 2019

Realizing the Dream

- So our model learns what it needs for all of these tasks, but how do we get it to do those tasks?

- Fine-tuning: e.g., train a simple MLP classifier on top of BERT sentence embeddings → immediate SOTA on many NLP tasks

- For autoregressive models:
  - Template-based prompting
  - Few-shot prompting

Template-Based Prompting

Main idea: format input to the model as if it were generic web text data.

Prompt:

```
Review: Let there be no question: Alexions owns the best cheeseburger
in the region and they have now for decades. Try a burger on Italian
bread. The service is flawlessly friendly, the food is amazing, and the
wings? Oh the wings... but it's still about the cheeseburger. The
atmosphere is inviting, but you can't eat atmosphere... so go right
now. Grab the car keys... you know you're hungry for an amazing
cheeseburger, maybe some wings, and a cold beer! Easily, hands down,
the best bar and grill in Pittsburgh.

On a 1 to 4 star scale, the reviewer would probably give this
restaurant a
```

Model's Output Probabilities:

P("1" | prompt) = 0.0016
P("2" | prompt) = 0.0032
P("3" | prompt) = 0.012
P("4" | prompt) = 0.78

Template used:

```
Review: <review>
On a 1 to 4 star scale, the reviewer would
probably give this restaurant a <class_label>
```

This template is tried for all possible class labels.

Classification (sentiment)

Template-Based Prompting

Main idea: format input to the model as if it were generic web text data.

Prompt:

```
India's moon rover completes its walk. Scientists analyzing data
looking for signs of frozen water

BEW DELHI -- India's moon rover has completed its walk on the lunar
surface and been put into sleep mode less than two weeks after its
historic landing near the lunar south pole, India's space mission said.

"The rover completes its assignments. It is now safely parked and set
into sleep mode," with daylight on that part of the moon coming to an
end, the Indian Space Research Organization said in a statement late
Saturday.

The rover's payloads are turned off and the data it collected has been
transmitted to the Earth via the lander, the statement said.

[...]

Active since the 1960s, India has launched satellites for itself and
other countries, and successfully put one in orbit around Mars in 2014.
India is planning its first mission to the International Space Station
next year, in collaboration with the United States.

TL;DR;
```

Model Generation:

```
India's moon rover has completed its assignments and gone to sleep mode
after just two weeks of being on the lunar surface. The rover has
started transmitting the data it has collected to Earth, but scientists
are unsure if it can survive the extremely cold temperatures on the
moon and wake up in order to conduct another set of missions when the
sun rises again. India successfully landed the rover and underscored
its status as a major tech power and space program.
```

Template used:

```
<title>

<article>

TL;DR;
```

We return the model's generation from the filled out template.

Text generation (summarization)

Example from CMU LLMs course

Few-Shot Prompting

Main idea: give lots of examples of the task in a consistent format.

Prompt:

```
The dog chased a squirrel at the park. = 那只狗在公园里追一只松鼠。

I was late for class. = 我上课迟到了。

The hippopotamus ate my homework. =
```

Model Generation:

河马吃了我的家庭作业。

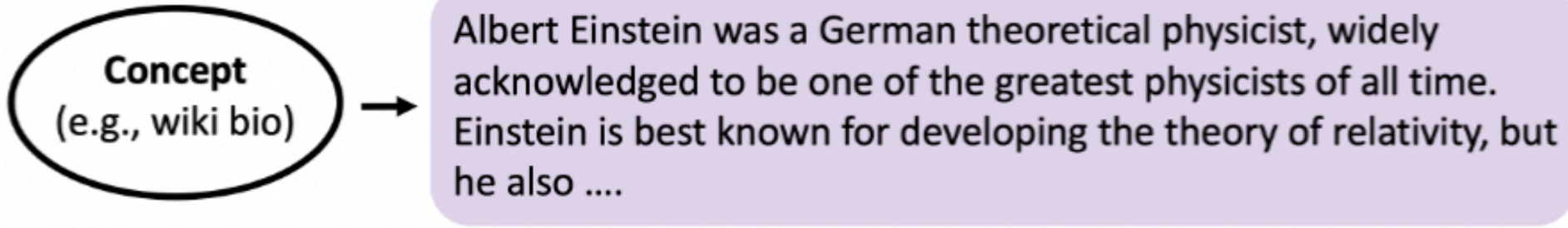
Template Used:

```
<example1_en> = <example1_zh>

<example2_en> = <example2_zh>

<query_en> =
```

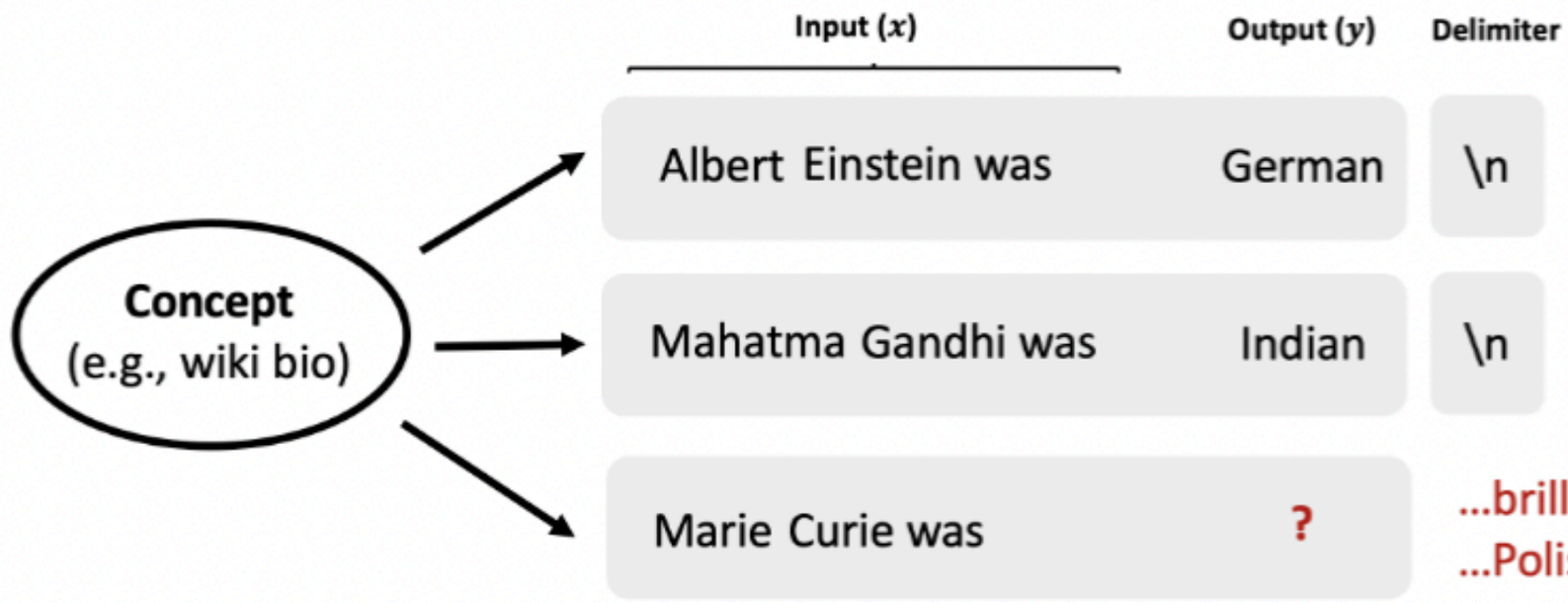
Text generation (translation)

Why does this work?

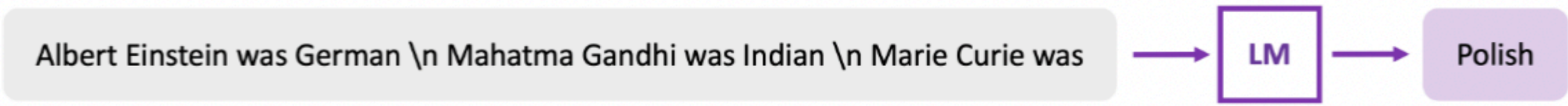
1. Pretraining documents are conditioned on a **latent concept** (e.g., biographical text)



2. Create **independent examples** from a **shared concept.** If we focus on full names, wiki bios tend to relate them to nationalities.

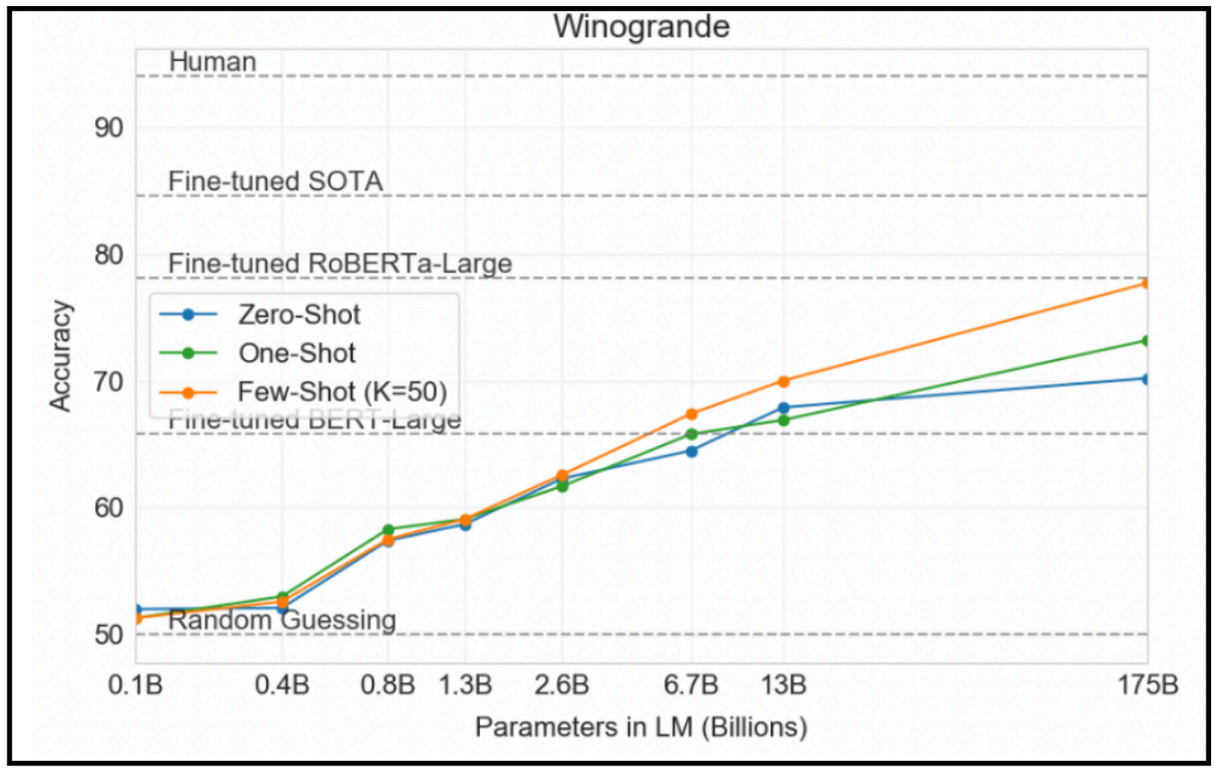


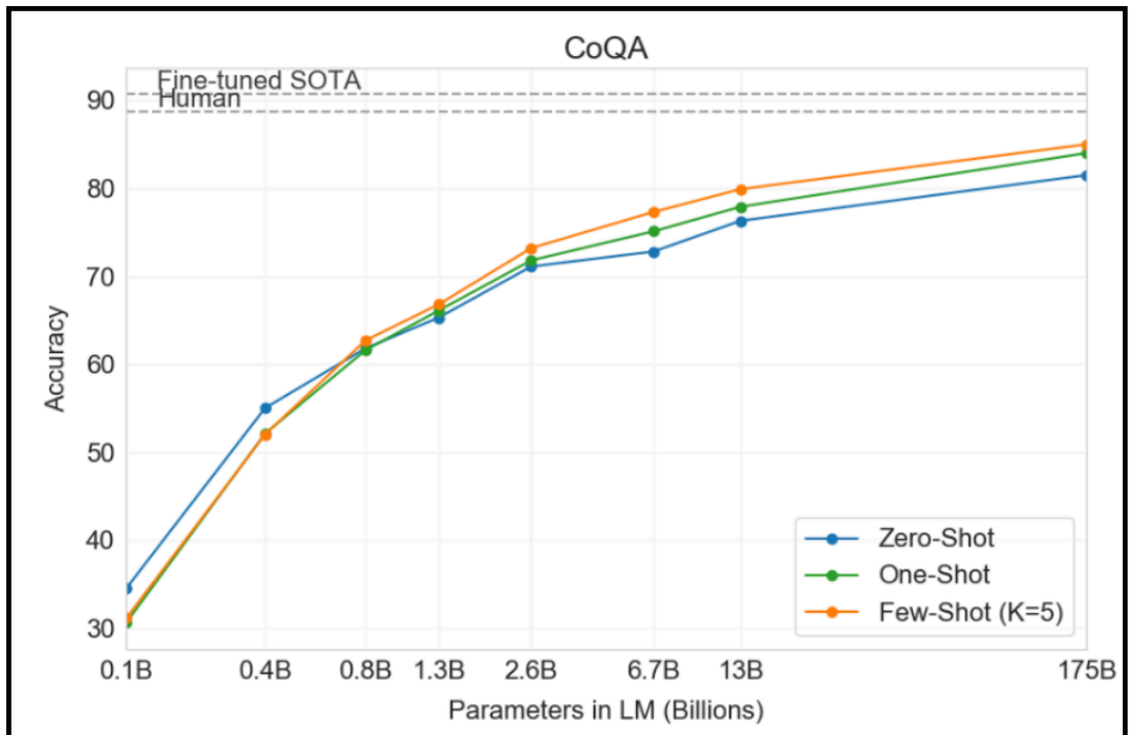
3. Concatenate examples into a prompt and predict next word(s). **Language model (LM) implicitly infers the shared concept** across examples despite the unnatural concatenation

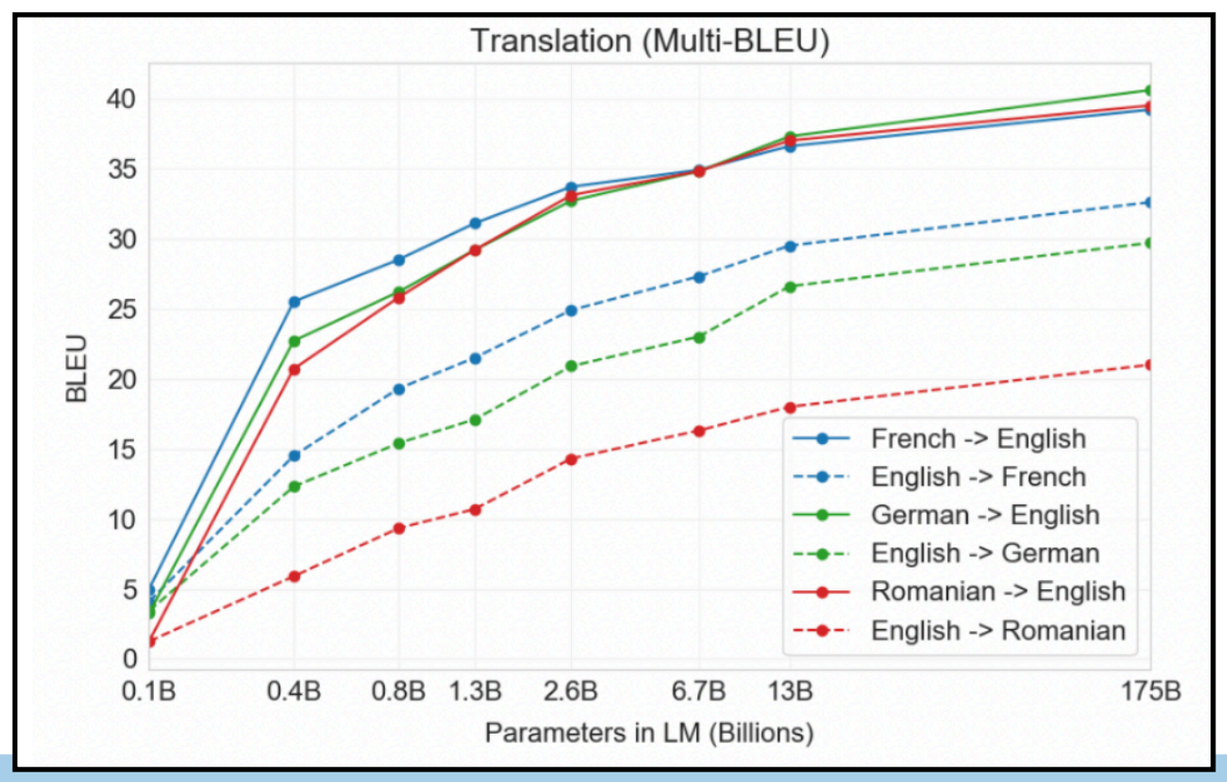


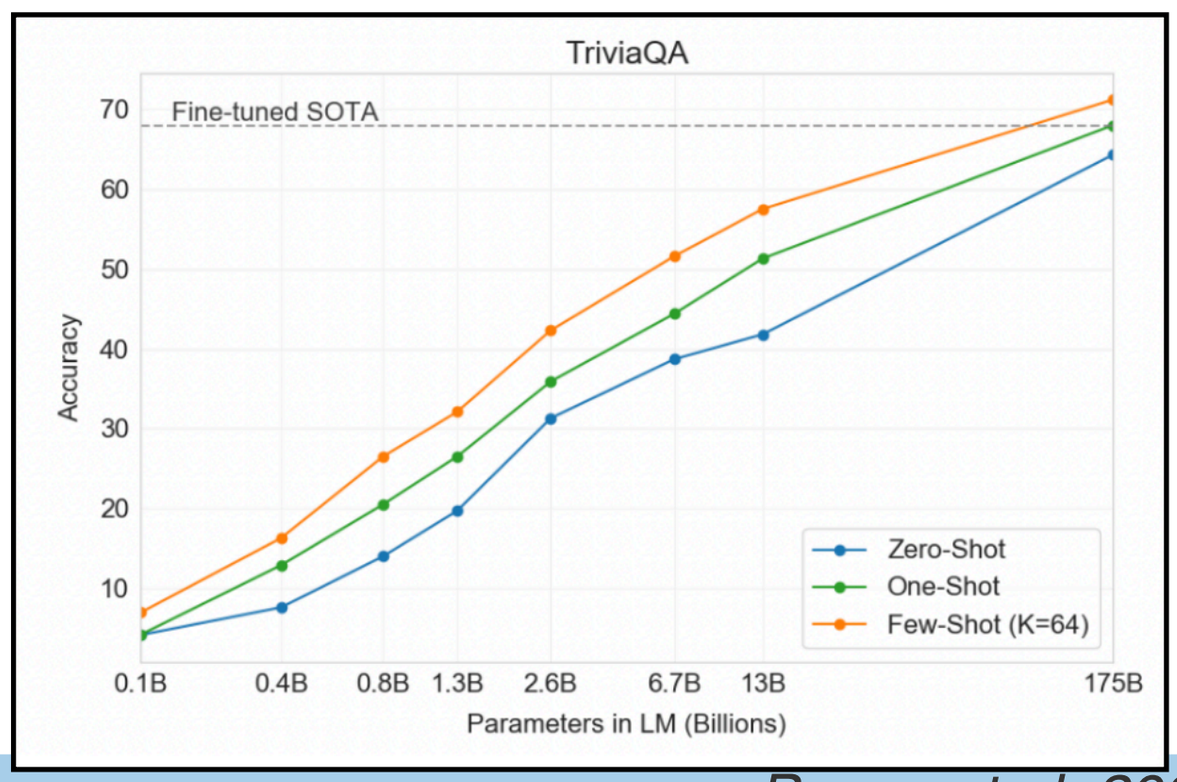
Xie et al. 2021

Realizing the Dream









Brown et al. 2020

Summary

- Traditional NLP was focused on improving on individual tasks, with the assumption we need to solve all of them before building "general" language systems

- But the answer was hiding inside us all along… language modeling gives us more than we thought!

- As models get better at language modeling, they get better at downstream tasks we care about
  - And we can show this by looking at their internal relationships

- We just need to figure out how to get them to **do** the tasks…