LLMs

Pretraining + Posttraining

Tatsunori Hashimoto

Roadmap

1. Pretraining (continued - Scaling)

2. Post-training

3. Statistical ML + LLMs

Pretraining through language modeling [Dai and Le, 2015]

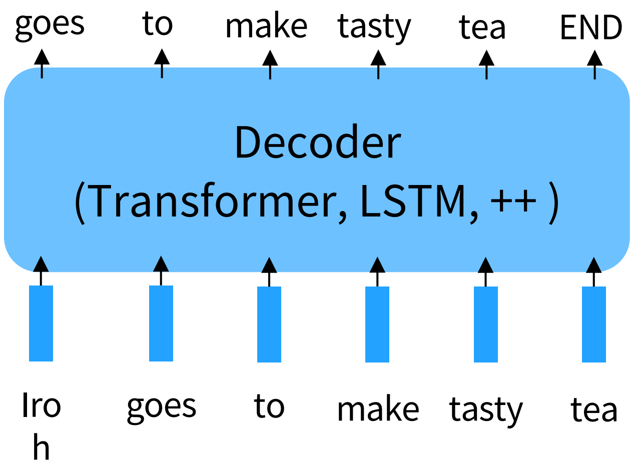
Recall the **language modeling** task:

- Model $p_\theta(w_t | w_{1:t-1})$, the probability distribution over words given their past contexts.
- There's lots of data for this! (In English.)

Pretraining through language modeling:

Train a neural network to perform language modeling on a large amount of text.

Save the network parameters.



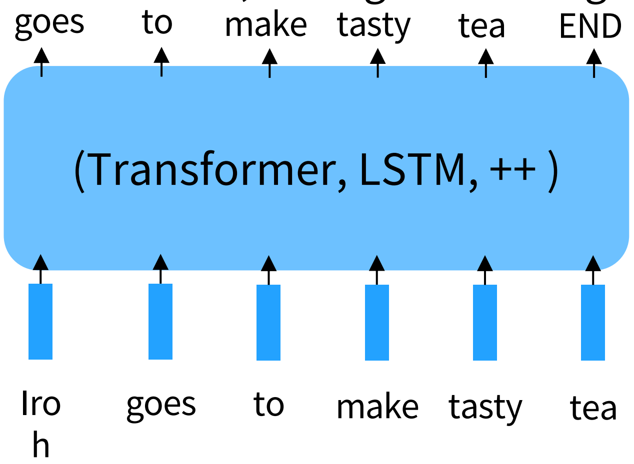
[Slide from CS224n]

The Pretraining / Finetuning Paradigm

Pretraining can improve NLP applications by serving as parameter initialization.

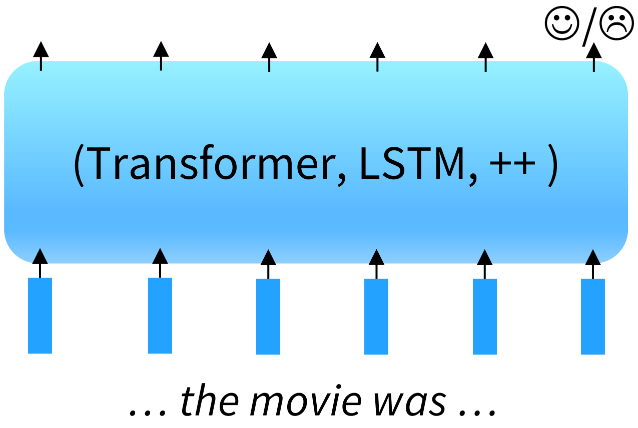
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



Step 2: Finetune (on your task)

Not many labels; adapt to the task!



[Slide from CS224n]

Stochastic gradient descent and pretrain/finetune

Why should pretraining and finetuning help, from a "training neural nets" perspective?

- Consider, provides parameters $\hat{\theta}$ by approximating $\min_{\theta} \mathcal{L}_{\text{pretrain}}(\theta)$.

  (The pretraining loss.)

- Then, finetuning approximates $\min_{\theta} \mathcal{L}_{\text{finetune}}(\theta)$, starting at $\hat{\theta}$.

  (The finetuning loss)

- The pretraining may matter because stochastic gradient descent sticks (relatively) close to $\hat{\theta}$ during finetuning.

  So, maybe the finetuning local minima near $\hat{\theta}$ tend to generalize well!

  And/or, maybe the gradients of finetuning loss near $\hat{\theta}$ propagate nicely!

[Slide from CS224n]

Pretraining decoders

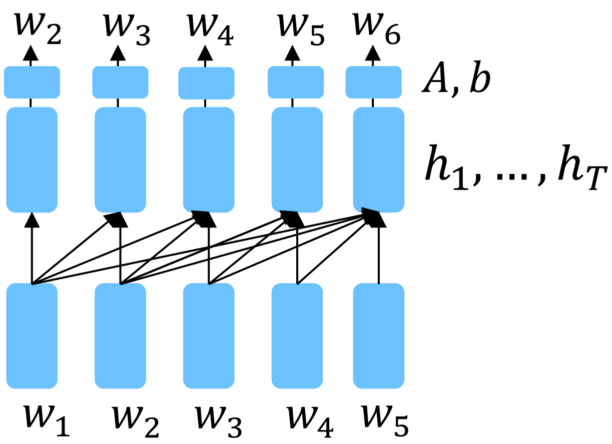
It's natural to pretrain decoders as language models and then use them as generators, finetuning their $p_\theta(w_t|w_{1:t-1})$!

This is helpful in tasks **where the output is a sequence** with a vocabulary like that at pretraining time!

- Dialogue (context=dialogue history)
- Summarization (context=document)

$$h_1, \dots, h_T = \text{Decoder}(w_1, \dots, w_T)$$
$$w_t \sim A h_{t-1} + b$$

Where A, b were pretrained in the language model!



[Note how the linear layer has been pretrained.]

Increasingly convincing generations (GPT2) [Radford et al., 2018]

Pretrained decoders can be used **in their capacities as language models.**

GPT-2, a larger version (1.5B) of GPT trained on more data, was shown to produce relatively convincing samples of natural language.

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.

[Slide from CS224n]

GPT-3, In-context learning, and very large models

Circa 2010s – we interact with LMs in two ways:

Sample from the distributions they define (maybe providing a prompt)

Fine-tune them on a task we care about, and take their predictions.

Emerging idea: leverage these models *without* explicit parameter adaptation

Very large language models seem to perform some kind of learning **without gradient steps** simply from examples you provide within their contexts.

GPT-3 is the canonical example of this. The largest T5 model had 11 billion parameters.

GPT-3 has 175 billion parameters.

[Slide from CS224n]

GPT-3, In-context learning, and very large models

Very large language models seem to perform some kind of learning **without gradient steps** simply from examples you provide within their contexts.

The in-context examples seem to specify the task to be performed, and the conditional distribution mocks performing the task to a certain extent.

Input (prefix within a single Transformer decoder context):

“ thanks -> merci

hello -> bonjour

mint -> menthe

otter -> ”

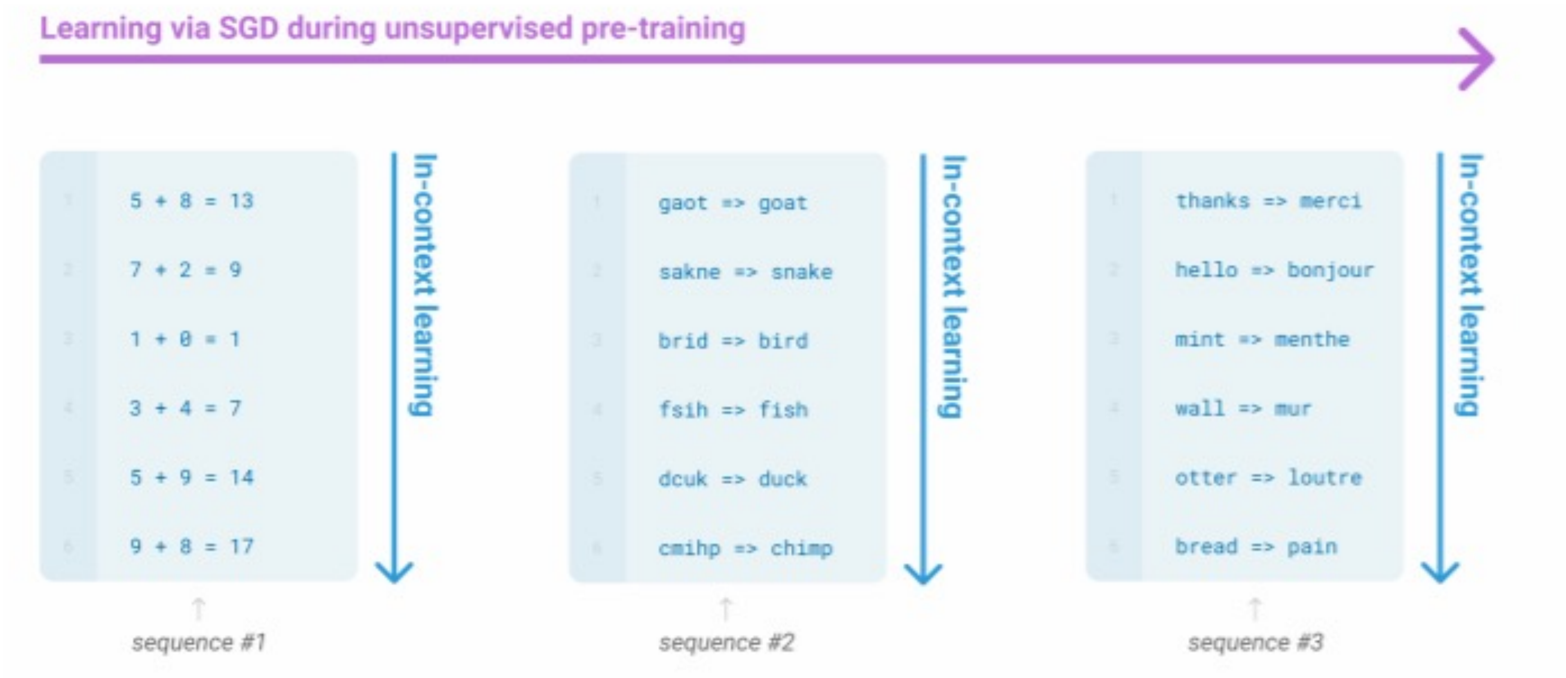
Output (conditional generations):

loutre…”

[Slide from CS224n]

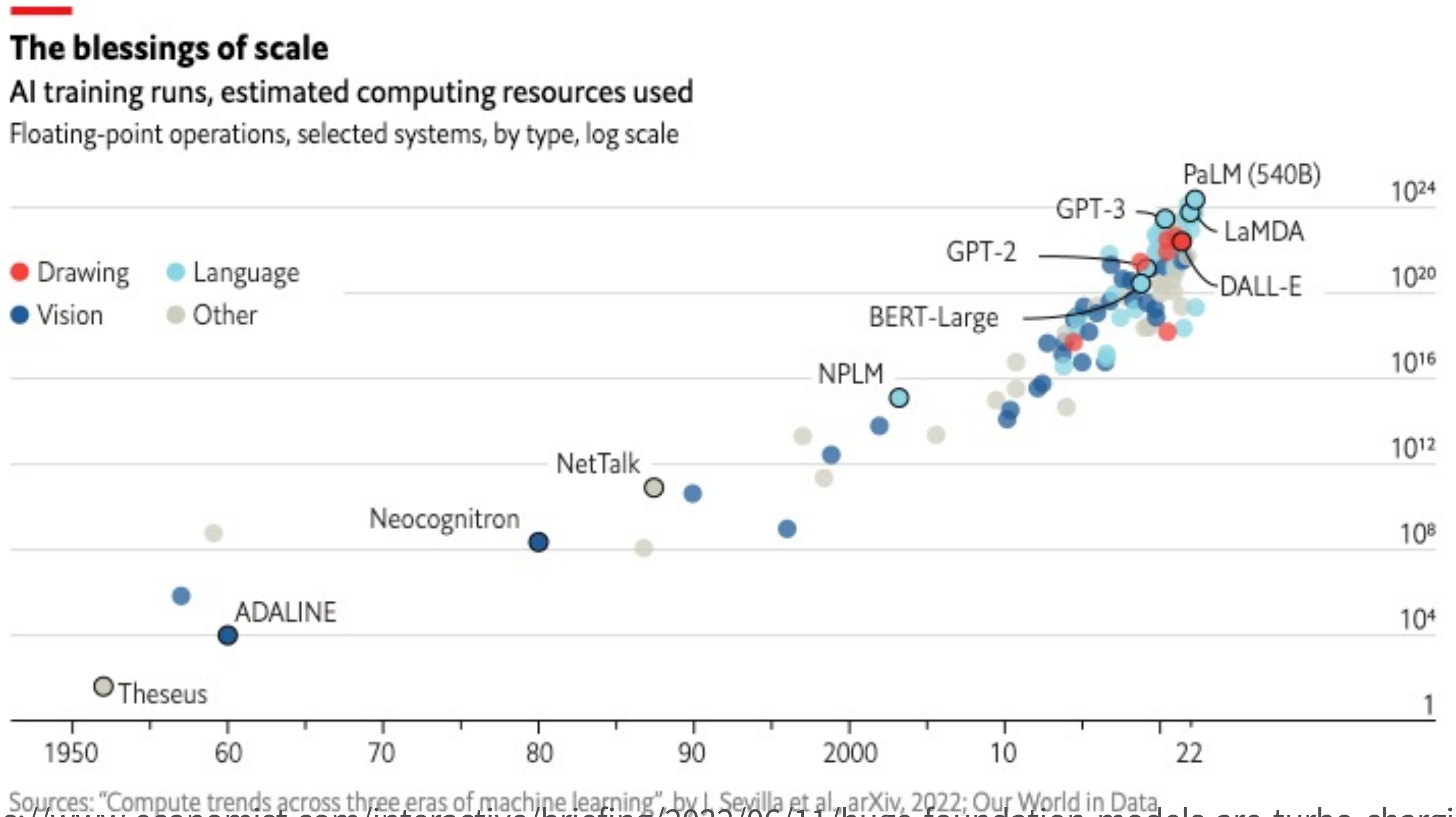
GPT-3, In-context learning, and very large models

Very large language models seem to perform some kind of learning **without gradient steps** simply from examples you provide within their contexts.



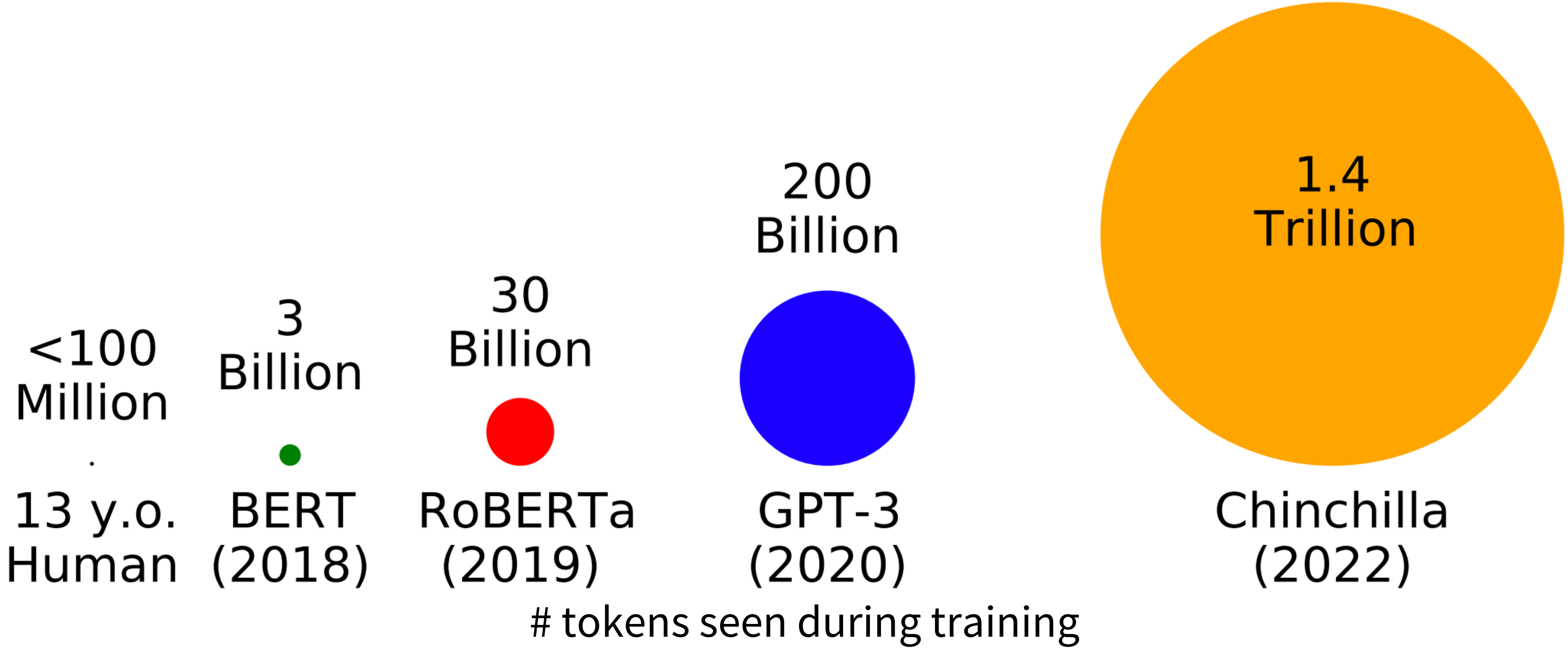
[Slide from CS224n]

Larger and larger models



https://www.economist.com/interactive/briefing/2022/06/11/huge-foundation-models-are-turbo-charging-ai-progress

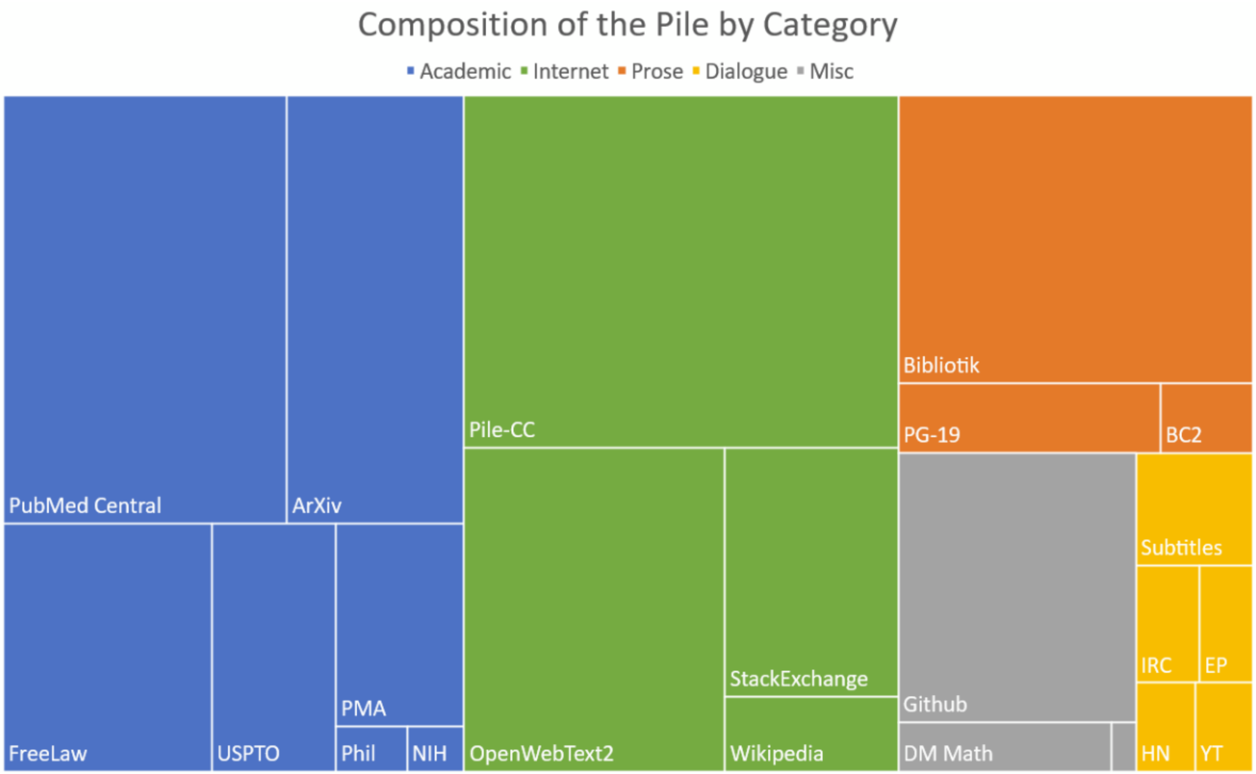
Trained on more and more data



https://babylm.github.io/

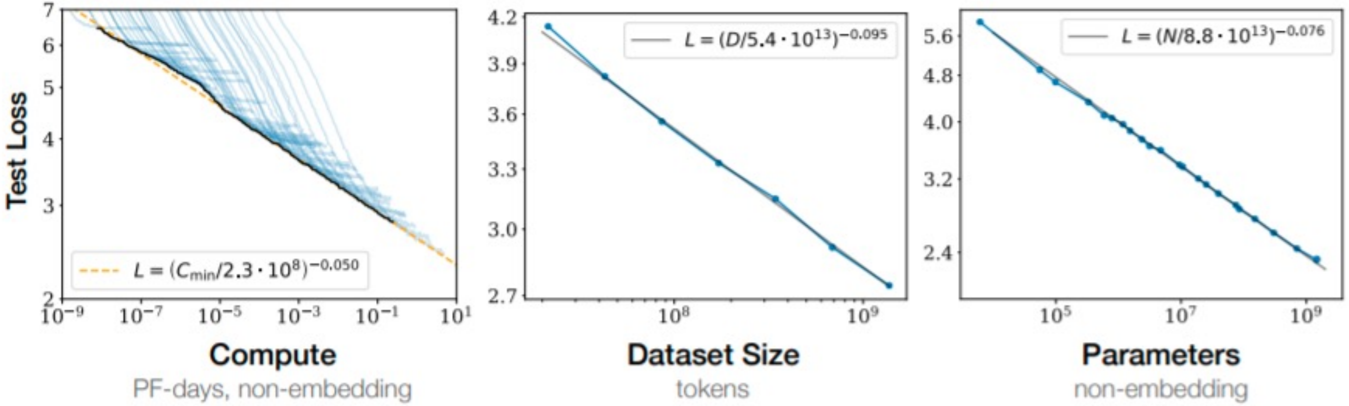
[Slide from CS224n]

Where does this data come from?



Model	Training Data
BERT	BookCorpus, English Wikipedia
GPT-1	BookCorpus
GPT-3	CommonCrawl, WebText, English Wikipedia, and 2 book databases (“Books 1” and “Books 2”)
GPT-3.5+	Undisclosed

Why scale? Scaling laws



Empirical observation: scaling up models leads to reliable gains in perplexity

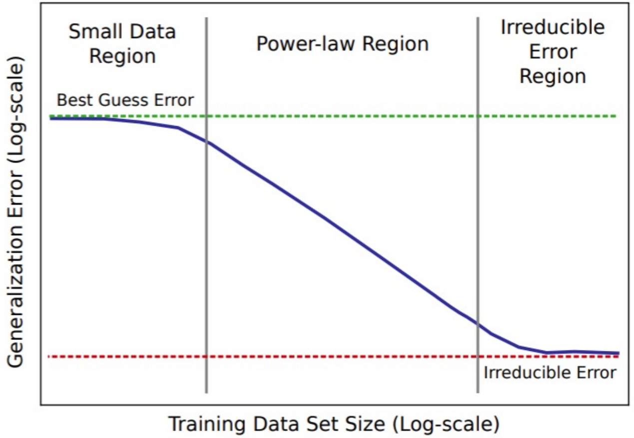
Data vs performance

What's a data scaling law?

Data scaling laws : simple formula that maps dataset size (n) to error

What do we expect out of scaling laws?

Monotonic, logistic-like curves

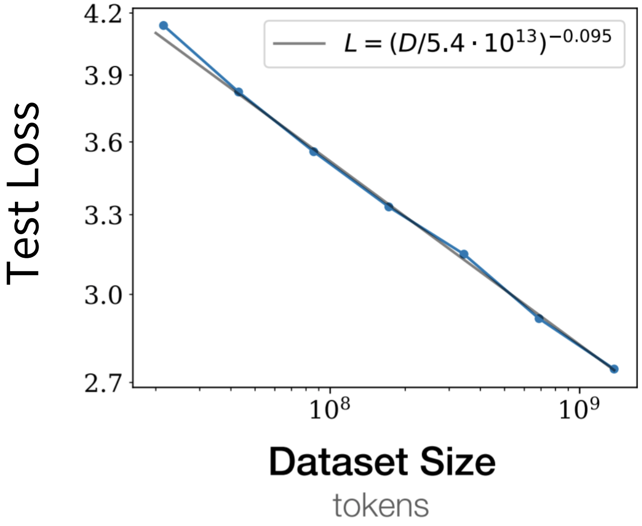


[Hestness+ 2017]

Data scaling laws for language models

First, an empirical observation

Loss and dataset size is linear on a log-log plot

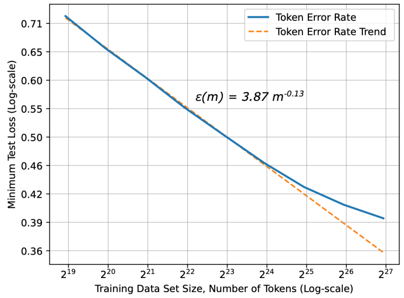


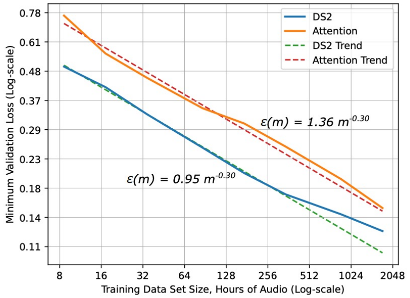
"Scale-free" or
"Power law"

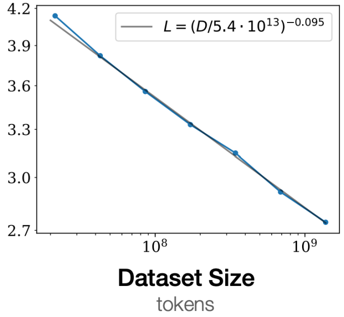
(For language modeling, from Kaplan+ 2020)

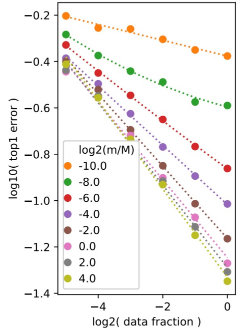
Scaling laws: past works and other areas

Scaling laws hold in many domains









Machine translation

Speech

Language modeling

Object recognition

Hestness et al 2017.

Kaplan et al 2020.

Rosenfeld 2020.

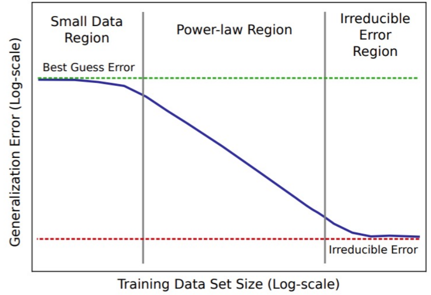
Data scaling has been known for a while

Kolachina+ 2012 for machine translation, Hestness+ 2017 for neural

Conceptual foundations of data scaling laws.

Q: Why do scaling laws show up?

We know error should be monotone



But why is it a power law / linear in log-log?

A: Estimation error naturally decays polynomially.

But this answer may take a moment to understand. Let's work through an example.

Example: If our task is to estimate the mean of a dataset, what's the scaling law?

Toy example: mean estimation

Input: $x_1 \; \ldots x_n \; \sim N(\mu, \sigma^2)$

Task: estimate the average as $\hat{\mu} = \frac{\sum_i x_i}{n}$

What's the error? By standard arguments..

$$\mathrm{E}[(\hat{\mu} - \mu)^2] = \frac{\sigma^2}{n}$$

This is a sample complexity bound and also a scaling law

$$\log(Error) = -\log n + 2 \log \sigma$$

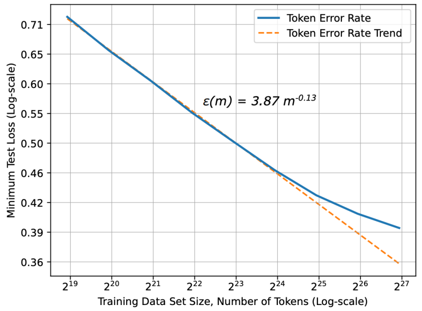
More generally, any polynomial rate $1/n^\alpha$ is a scaling law

Scaling law exponents: an intriguing mystery

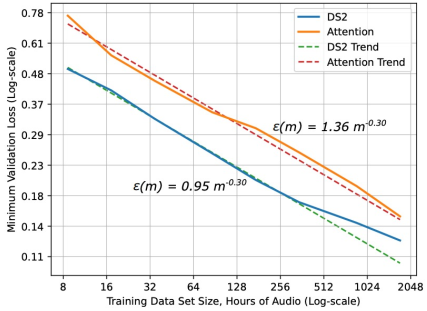
Similar arguments show most 'classical' models (regression, etc) have $\frac{1}{n}$ or $\frac{1}{\sqrt{n}}$ scaling

This means we should see $y = -x + C$ or $y = -0.5x + C$

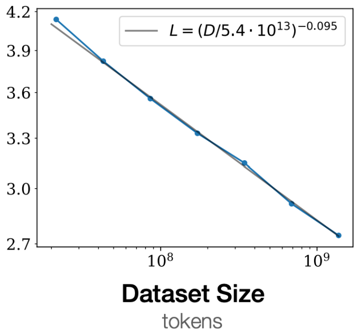
What do we find in neural scaling laws?



Machine translation



Speech



Language modeling

Why? – still somewhat an open question

Detour: scaling laws for (nonparametric) learning

Neural nets can approximate arbitrary functions. Lets turn that into an example.

Input: $x_1 \ldots x_n$ uniform in 2D unit box. $y_i = f(x_i) + N(0,1)$

Task: estimate f(x)

Approach: cut up the 2D space into boxes with length $n^{-\frac{1}{4}}$, average in each box

What's our estimation error?

Informally, we have $\sqrt{n}$ boxes, each box gets $\sqrt{n}$ samples.

$$Error \approx \frac{1}{\sqrt{n}} + (other\ smoothness\ terms)$$

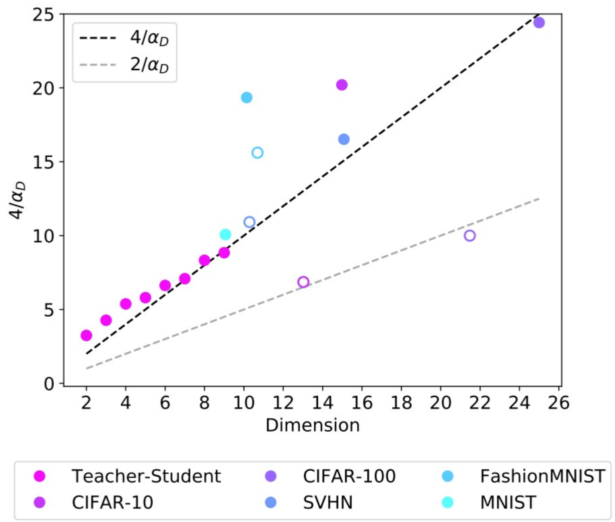
In d-dimensions, this becomes $Error = n^{-1/d}$ - **This means scaling is** $\boldsymbol{y = -\frac{1}{d}x + C}$

Takeaway: flexible 'nonparametric' learning has dimension dependent scaling laws.

Intrinsic dimensionality theory of data scaling laws

In case that was a bit too low-level..

1. Scaling laws arise due to polynomial rates of learning $\frac{1}{n^{\alpha}}$
2. Some argue the slope α is closely connected to the *intrinsic dimensionality* of the data.



Some recent work (Bahri+ 2021) have tried to verify this empirically.. though no smoking gun

Scaling laws for model engineering

Scaling underlies many ‘systematic’ design choices for LMs

Our motivation: how can we efficiently design huge LMs?

- LSTMs vs Transformers
- Adam vs SGD

How should we allocate our limited resources?

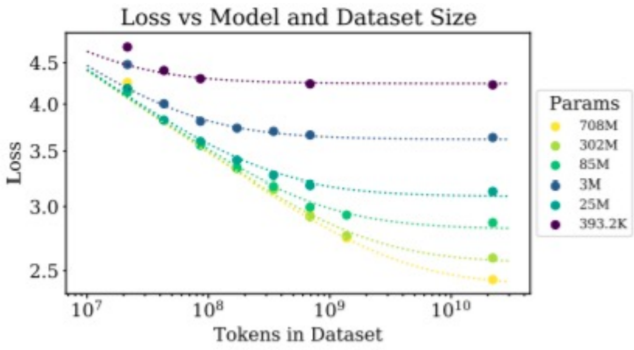
- Train models longer vs train bigger models?
- Collect more data vs get more GPUs?

Scaling laws provide a simple procedure to answer these.

Model size data joint scaling

Q: Do we need more data or bigger models?

Clearly, lots of data is wasted on small models



Joint data-model scaling laws describe how the two relate

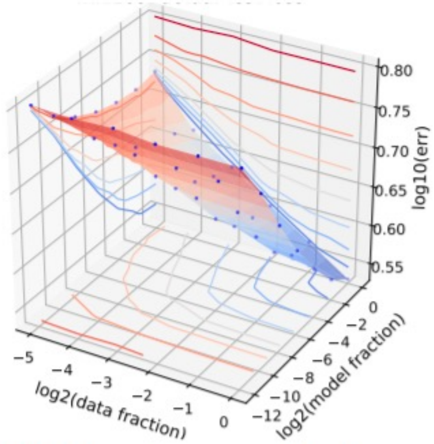
From Rosenfeld+ 2020,

$$Error = n^{-\alpha} + m^{-\beta} + C$$

From Kaplan+ 2021

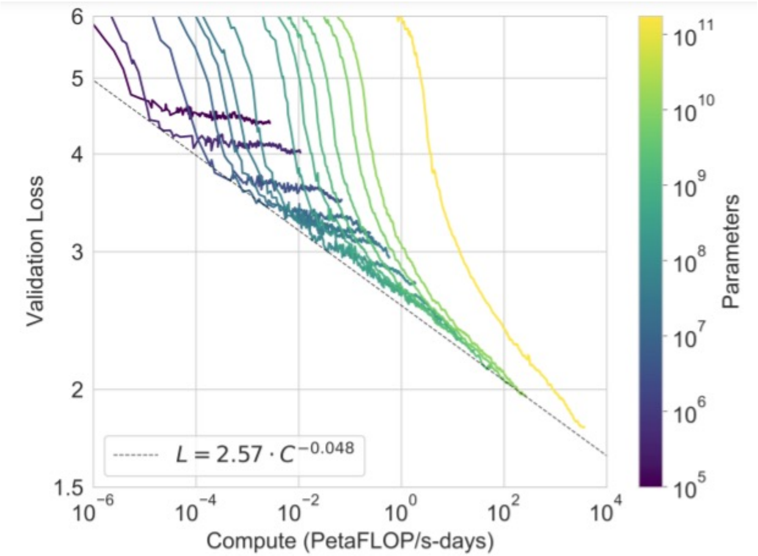
$$Error = [m^{-\alpha} + n^{-1}]^{\beta}$$

Provides surprisingly good fits to model-data joint error.



(a) Wiki103 error (cross entropy) landscape.

Scaling can help identify model size – data tradeoffs

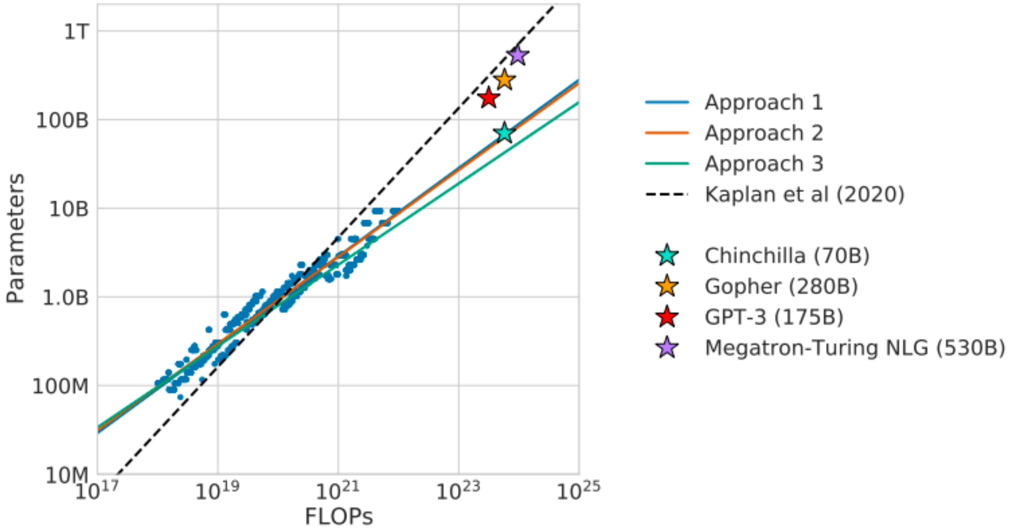


Modern observation: train a big model that's not fully converged.

Caution #2 – ‘Optimal’ scaling laws are hard to get

‘Linear fits’ can be quite deceiving

Minor changes to hyperparameters result in different scaling



Hoffman+ 2022

Scaling Efficiency: how do we best use our compute

GPT-3 was **175B parameters** and trained on **300B** tokens of text.

Roughly, the cost of training a large transformer scales as **parameters*tokens**

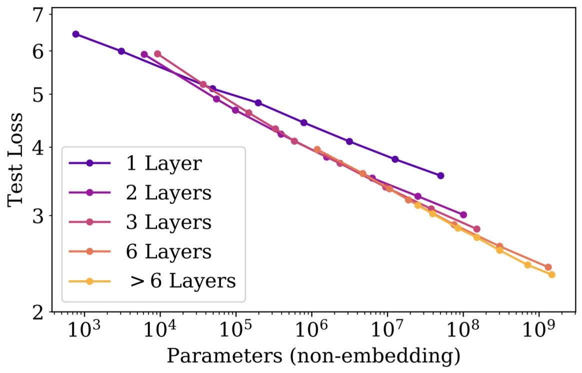
Did OpenAI strike the right parameter-token data to get the best model? No.

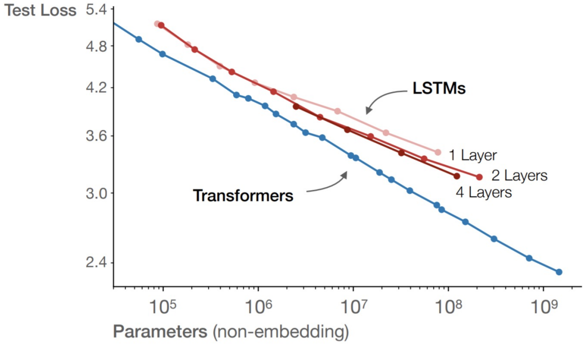
Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion

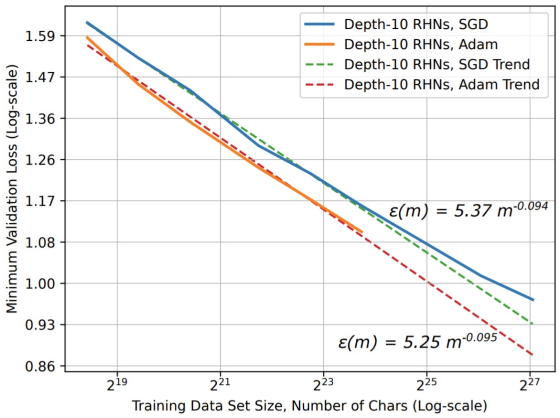
This 70B parameter model is better than the much larger other models!

[Slide from CS224n]

Scaling laws for many other interesting architecture decisions







Predictable scaling helps us make intelligent decisions about architectures etc.

Scaling laws for models and compute

Log-linearity extends to model parameters and compute!

Lets us set the following based on small models

- Pick optimizer
- Pick architecture and model sizes

Also lets us make smart resource tradeoffs

- Big models vs more data?

Language models as world models?

Language models may do rudimentary modeling of *agents*, *beliefs*, and *actions*:

> ***Pat watches a demonstration of a bowling ball and a leaf being dropped at the same time in a vacuum chamber. Pat, who is a physicist, predicts that*** *the bowling ball and the leaf will fall at the same rate.*

Changing the last sentence of the prompt, we get:

> ***. . . Pat, who has never seen this demonstration before, predicts that*** *the bowling ball will fall to the ground first. This is incorrect. In a vacuum chamber, there is no air*

Language Models as Agent Models [Andreas, 2022]

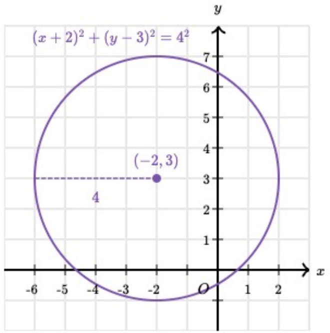
[Slide from CS224n]

Language models as world models?

…*math*:

We can describe circles in the xy-plane using equations in terms of x and y. Circle equations questions require us to understand the connection between these equations and the features of circles.

For example, the equation $(x+2)^2+(y-3)^2=4^2$ is graphed in the xy-plane below. It is a circle with a center at $(-2,3)$ and a radius of 4.



https://www.khanacademy.org/test-prep/sat/x0a8c2e5f:untitled-652

[Slide from CS224n]

Language models as world models?

…*code*:

```
// Determine whether the sentiment of text is positive
// Use a web service
async function isPositive(text: string): Promise<boolean> {
  const response = await fetch(`http://text-processing.com/api/sentiment/`, {
    method: "POST",
    body: `text=${text}`,
    headers: {
      "Content-Type": "application/x-www-form-urlencoded",
    },
  });
  const json = await response.json();
  return json.label === "pos";
}
```

https://github.com/features/copilot

[Slide from CS224n]

Language models as world models?

…*medicine*:

Rapid and chronic ethanol tolerance are composed of distinct memory-like states in Drosophila

Abstract

Ethanol tolerance is the first type of behavioral plasticity and neural plasticity that is induced by ethanol intake, and yet its molecular and circuit bases remain largely unexplored. Here, we characterize three distinct forms of ethanol tolerance in male Drosophila: rapid, chronic, and repeated. Rapid tolerance is composed of two short-lived memory-like states, one that is labile and one that is consolidated. Chronic tolerance, induced by continuous exposure, lasts for two days, induces ethanol preference, and hinders the development of rapid tolerance through the activity of

[Larnerd, 2023]

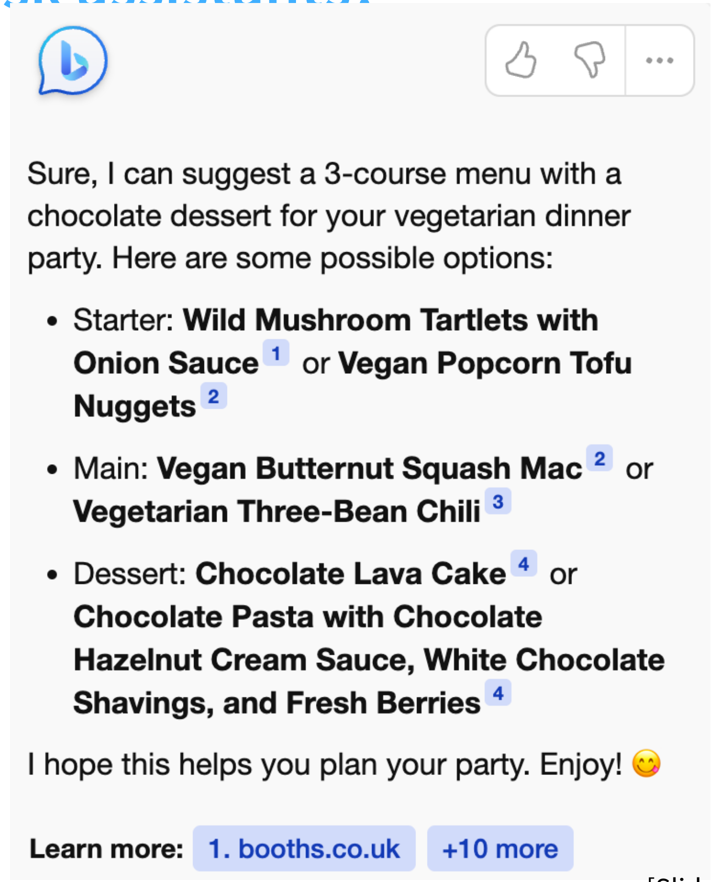
[Slide from CS224n]

Language models as multitask assistants?

I need to throw a dinner party for 6 people who are vegetarian. Can you suggest a 3-course menu with a chocolate dessert?

[Microsoft Bing]

(Also see OpenAI's ChatGPT,
Google's Bard, Anthropic's Claude)



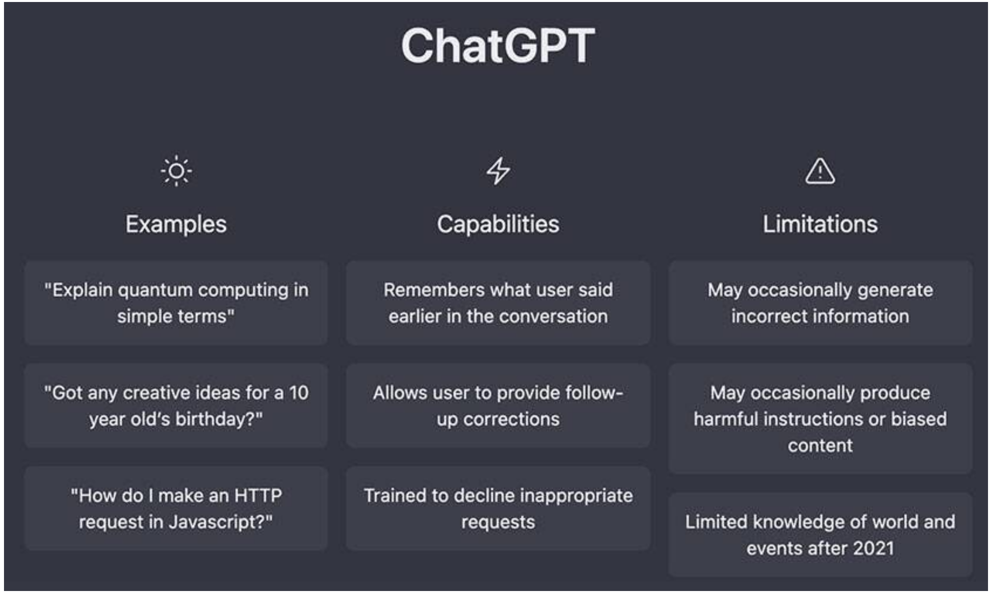
[Slide from CS224n]

Language models as multitask assistants?

How do we get from *this*

Stanford University is located in ___________

to *this*?



[Slide from CS224n]

Language modeling ≠ assisting users

PROMPT *Explain the moon landing to a 6 year old in a few sentences.*

COMPLETION GPT-3

```
Explain the theory of gravity to a 6 year old.

Explain the theory of relativity to a 6 year old in a few sentences.

Explain the big bang theory to a 6 year old.

Explain evolution to a 6 year old.
```

Language models are not *aligned* with user intent [Ouyang et al., 2022].

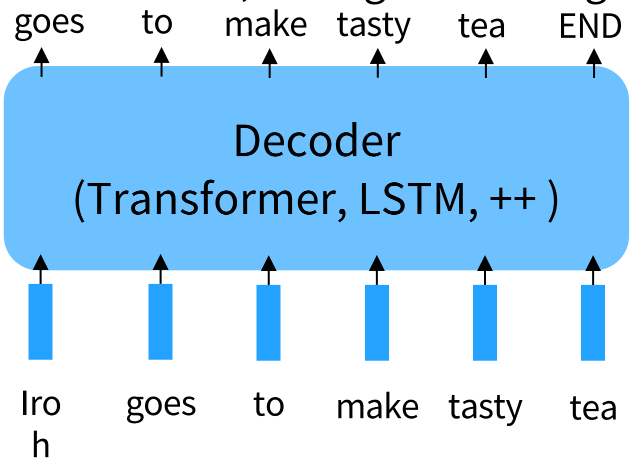
[Slide from CS224n]

Scaling up finetuning

Pretraining can improve NLP applications by serving as parameter initialization.

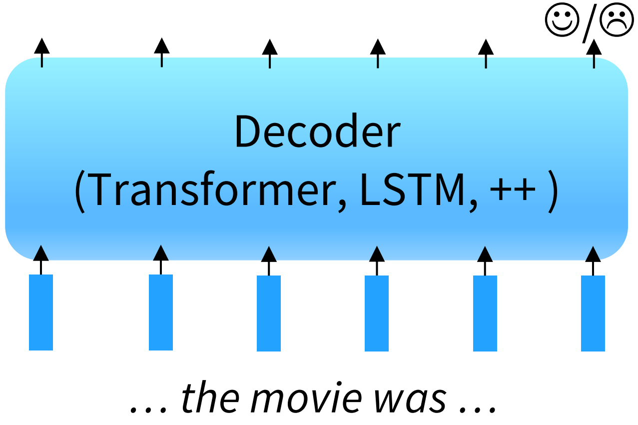
Step 1: Pretrain (on language modeling)

Lots of text; learn general things!



Step 2: Finetune (on many tasks)

~~Not~~ many labels; adapt to the tasks!

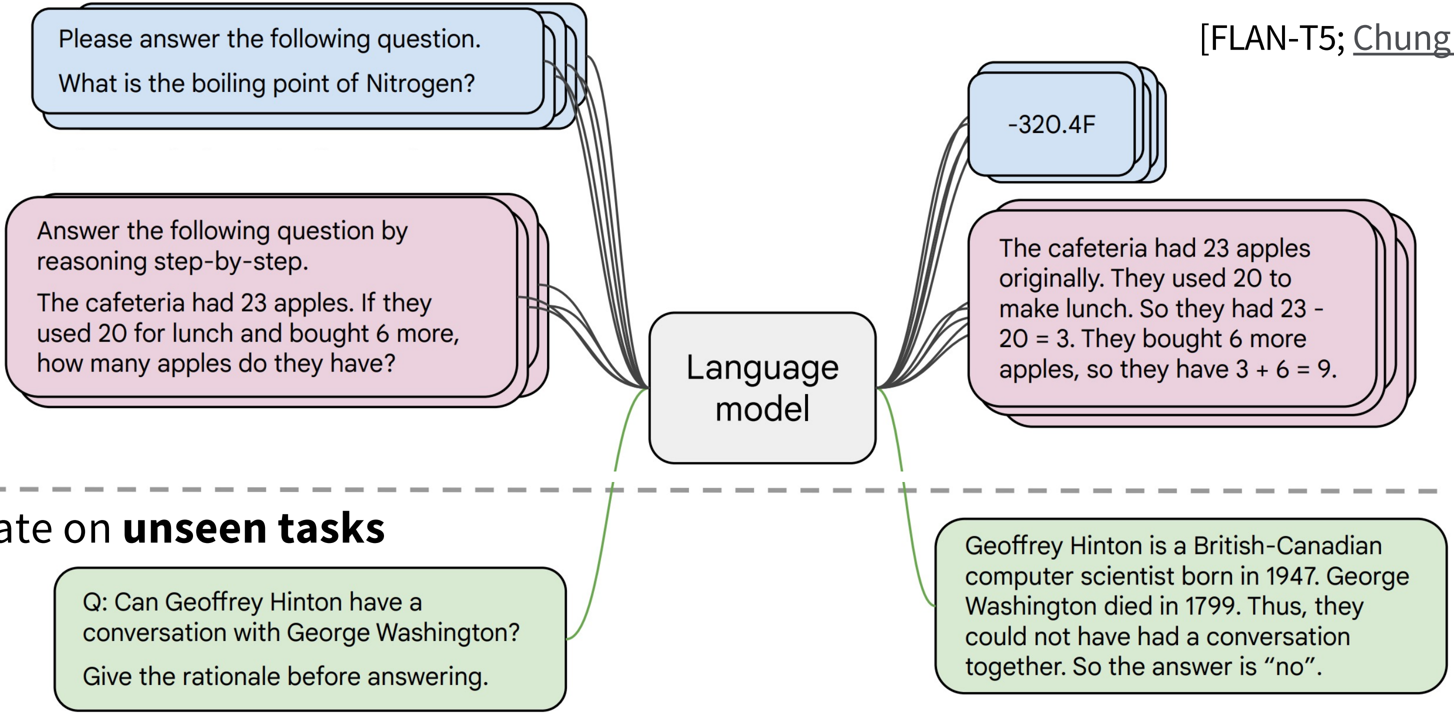


[Slide from CS224n]

Instruction finetuning

Collect examples of (instruction, output) pairs across many tasks and finetune an LM

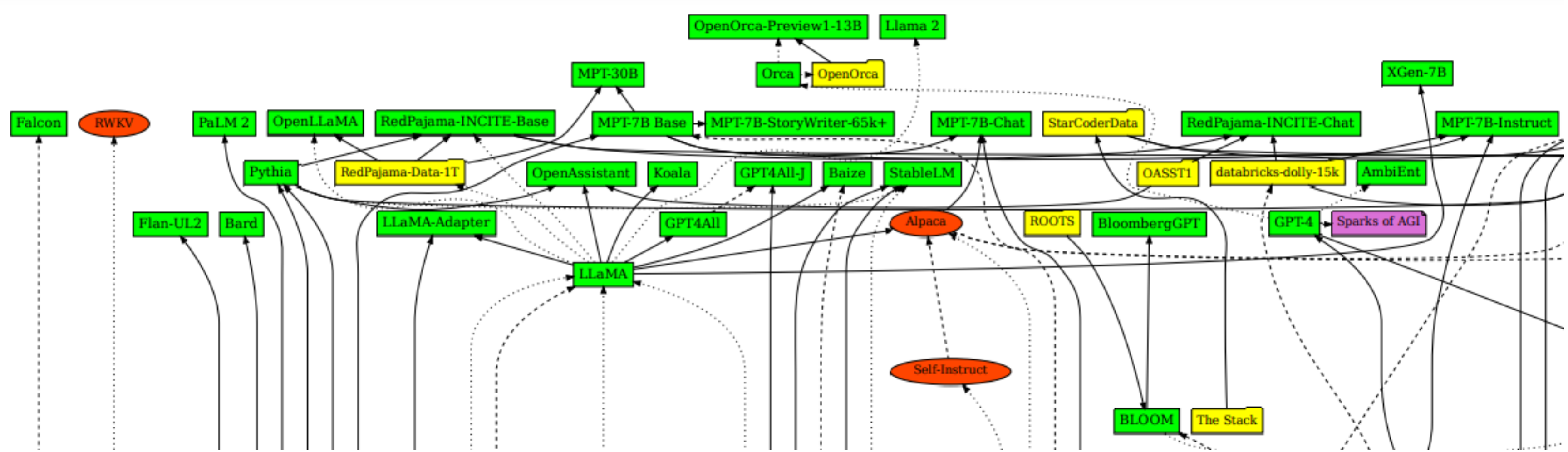
[FLAN-T5; Chung et al., 2022]



- Evaluate on **unseen tasks**

[Slide from CS224n]

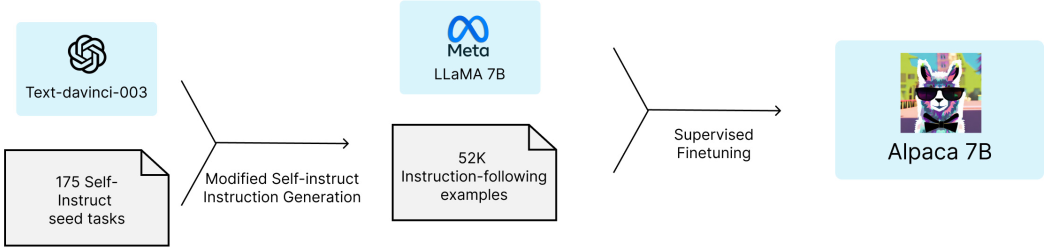
A huge diversity of instruction-tuning datasets



The release of LLaMA led to open-source attempts to `create’ instruction tuning data

What have we learned from this?

You can generate data synthetically (from bigger LMs)

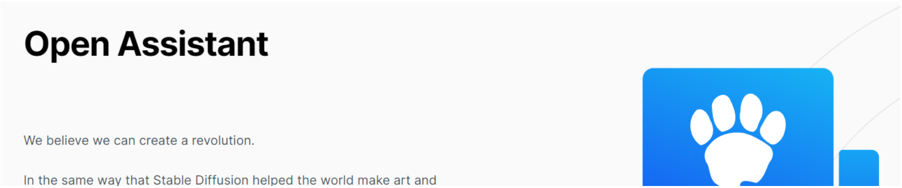


You don't need many samples to instruction tune


LIMA: Less Is More for Alignment

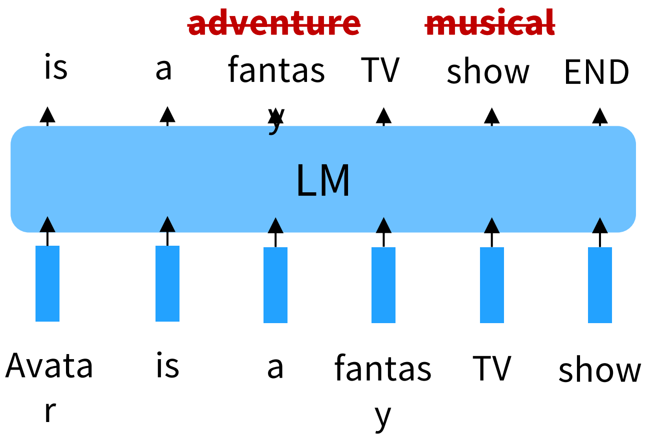
Chunting Zhou[μ*] Pengfei Liu[π*] Puxin Xu[μ] Srini Iyer[μ] Jiao Sun[λ]

Crowdsourcing can be pretty effective!



Limitations of instruction finetuning?

- One limitation of instruction finetuning is obvious: it's **expensive** to collect ground-truth data for tasks.
- But there are other, subtler limitations too. Can you think of any?
- **Problem 1:** tasks like open-ended creative generation have no right answer.
  - *Write me a story about a dog and her pet grasshopper.*
- **Problem 2:** language modeling penalizes all token-level mistakes equally, but some errors are worse than others.
- Even with instruction finetuning, there a mismatch between the LM objective and the objective of "satisfy human preferences"!
- Can we **explicitly attempt to satisfy human preferences?**



[Slide from CS224n]

Optimizing for human preferences

- Let's say we were training a language model on some task (e.g. summarization).
- For each LM sample s, imagine we had a way to obtain a *human reward* of that summary: $R(s) \in \mathbb{R}$, higher is better.

```
SAN FRANCISCO,
California (CNN) --
A magnitude 4.2
earthquake shook
the San Francisco
...
overturn unstable
objects.
```

```
An earthquake hit
San Francisco.
There was minor
property damage,
but no injuries.
```

s_1

$R(s_1) = 8.0$

```
The Bay Area has
good weather but
is prone to
earthquakes and
wildfires.
```

s_2

$R(s_2) = 1.2$

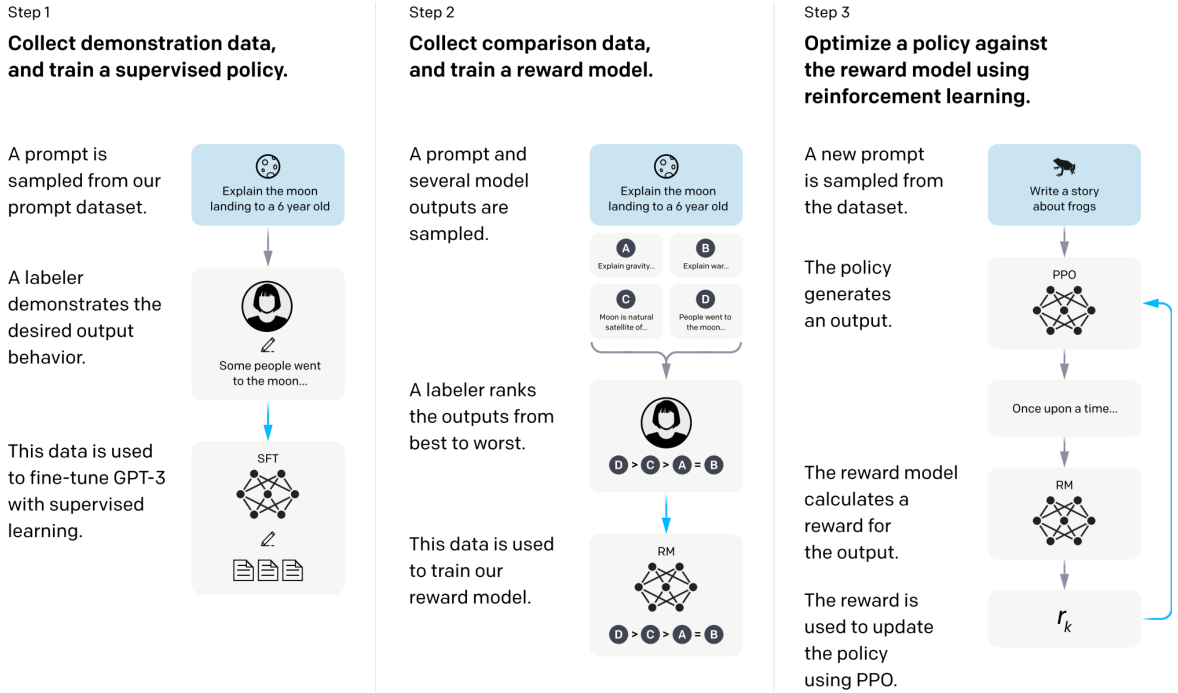
- Now we want to maximize the expected reward of samples from our LM:

$$\mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})]$$

Note: for mathematical simplicity we're assuming only one "prompt"

[Slide from CS224n]

High-level instantiation: ‘RLHF’ pipeline



First step: instruction tuning!

Second + third steps: maximize reward (but how??)

Optimizing for human preferences

- How do we actually change our LM parameters θ to maximize this?

$$\mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})]$$

- Let's try doing gradient ascent!

$$\theta_{t+1} := \theta_t + \alpha \nabla_{\theta_t} \mathbb{E}_{\hat{s}\sim p_{\theta_t}(s)}[R(\hat{s})]$$

How do we estimate this expectation??

What if our reward function is non-differentiable??

- **Policy gradient** methods in RL (e.g., REINFORCE; [Williams, 1992]) give us tools for estimating and optimizing this objective.
- We'll describe a *very high-level mathematical* overview of the simplest policy gradient estimator.

[Slide from CS224n]

A (very!) brief introduction to policy gradient/REINFORCE [Williams, 1992]

- We want to obtain

(defn. of expectation) (linearity of gradient)

$$\nabla_\theta \mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})] = \nabla_\theta \sum_s R(s) p_\theta(s) = \sum_s R(s)\, \nabla_\theta p_\theta(s)$$

- Here we'll use a very handy trick known as the **log-derivative trick**. Let's try taking the gradient of $\log p_\theta(s)$

$$\nabla_\theta \log p_\theta(s) = \frac{1}{p_\theta(s)} \nabla_\theta p_\theta(s) \qquad \Longrightarrow \qquad \nabla_\theta\, p_\theta(s) = p_\theta(s)\ \nabla_\theta \log p_\theta(s)$$

(chain rule)

- Plug back in:

This is an expectation of this

$$\sum_s R(s)\, \nabla_\theta p_\theta(s) = \sum_s p_\theta(s) R(s)\, \nabla_\theta \log p_\theta(s)$$

$$\boxed{= \mathbb{E}_{\hat{s}\sim p_\theta(s)}[R(\hat{s})\, \nabla_\theta \log p_\theta(\hat{s})]}$$

[Slide from CS224n]

How do we model human preferences?

- Awesome: now for any **arbitrary, non-differentiable reward function** $R(s)$, we can train our language model to maximize expected reward.
- Not so fast! (Why not?)
- **Problem 1:** human-in-the-loop is expensive!
  - **Solution:** instead of directly asking humans for preferences, **model their preferences** as a separate (NLP) problem! [Knox and Stone, 2009]

```
An earthquake hit
San Francisco.
There was minor
property damage,
but no injuries.
```

s_1

$R(s_1) = 8.0$

```
The Bay Area has
good weather but
is prone to
earthquakes and
wildfires.
```

s_2

$R(s_2) = 1.2$

Train an LM $RM_\phi(s)$ to predict human preferences from an annotated dataset, then optimize for RM_ϕ instead.

[Slide from CS224n]

How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable [Phelps et al., 2015; Clark et al., 2018]

```
A 4.2 magnitude
earthquake hit
San Francisco,
resulting in
massive damage.
```

$$s_3$$

$$R(s_3) = \ ?$$

$$R(s_3) = \quad 4.1? \quad 6.6? \quad 3.2?$$

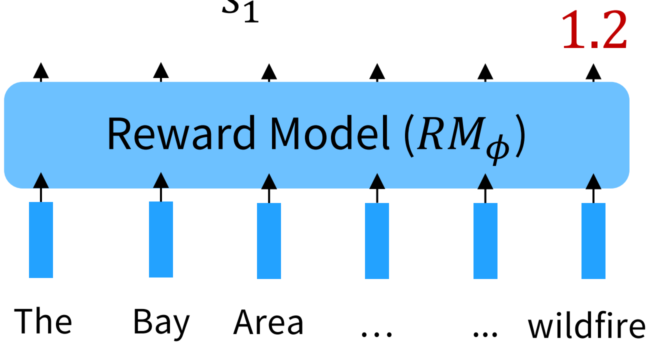
[Slide from CS224n]

How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable [Phelps et al., 2015; Clark et al., 2018]

An earthquake hit San Francisco. There was minor property damage, but no injuries. > A 4.2 magnitude earthquake hit San Francisco, resulting in massive damage. > The Bay Area has good weather but is prone to earthquakes and wildfires.

s_1 s_3 s_2



Bradley-Terry [1952] paired comparison model

$$J_{RM}(\phi) = -\mathbb{E}_{(s^w, s^l)\sim D}\left[\log \sigma(RM_\phi(s^w) - RM_\phi(s^l))\right]$$

"winning" sample "losing" sample

s^w should score higher than s^l

RLHF: Putting it all together [Christiano et al., 2017; Stiennon et al., 2020]

- Finally, we have everything we need:
  - A pretrained (possibly instruction-finetuned) LM $p^{PT}(s)$
  - A reward model $RM_\phi(s)$ that produces scalar rewards for LM outputs, trained on a dataset of human comparisons
  - A method for optimizing LM parameters towards an arbitrary reward function.
- Now to do RLHF:
  - Initialize a copy of the model $p_\theta^{RL}(s)$, with parameters θ we would like to optimize
  - Optimize the following reward with RL:

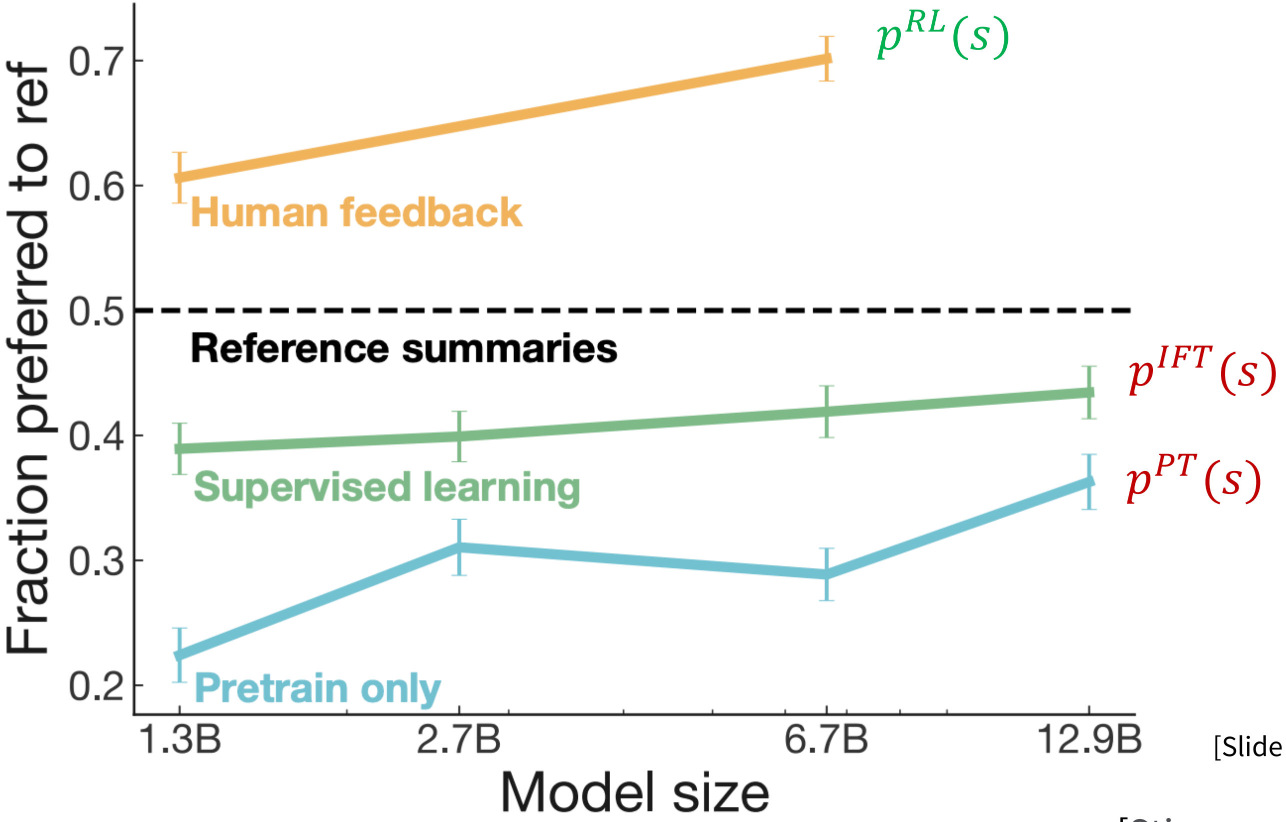
$$R(s) = RM_\phi(s) - \underbrace{\beta \log \left(\frac{p_\theta^{RL}(s)}{p^{PT}(s)} \right)}$$

Pay a price when $p_\theta^{RL}(s) > p^{PT}(s)$

[Slide from CS224n]

This is a penalty which prevents us from diverging too far from the pretrained model. In expectation, it is known as the **Kullback-Leibler** (**KL**) divergence between $p_\theta^{RL}(s)$ and $p^{PT}(s)$.

RLHF provides gains over pretraining + finetuning



[Slide from CS224n]

[Stiennon et al., 2020]

Controlled comparisons of “RLHF” style algorithms

Method	Simulated win-rate (%)	Human win-rate (%)
GPT-4	79.0 ± 1.4	69.8 ± 1.6
ChatGPT	61.4 ± 1.7	52.9 ± 1.7
PPO	46.8 ± 1.8	55.1 ± 1.7
Best-of-n	45.0 ± 1.7	50.7 ± 1.8
Expert Iteration	41.9 ± 1.7	45.7 ± 1.7
SFT 52k (Alpaca 7B)	39.2 ± 1.7	40.7 ± 1.7
SFT 10k	36.7 ± 1.7	44.3 ± 1.7
Binary FeedME	36.6 ± 1.7	37.9 ± 1.7
Quark	35.6 ± 1.7	-
Binary Reward Conditioning	32.4 ± 1.6	-
Davinci001	24.4 ± 1.5	32.5 ± 1.6
LLaMA 7B	11.3 ± 1.1	6.5 ± 0.9

Many works study RLHF behaviors using GPT-4 feedback (**Simulated**) as a surrogate for **Human** feedback.

PPO (method in InstructGPT) does work

Simple baselines (Best-of-n, Training on ‘good’ outputs) works well too

[Dubois et al 2023]

RLHF behaviors – clear stylistic changes

Instruction: What are the five most common causes of stress among people?

Alpaca (Finetuned w/ 10k data): The five most common causes of stress among people are work, money, relationships, health, and family

Alpaca + PPO: The five most common causes of stress among people are:

1. Pressure at work: Whether it is due to tight deadlines, high expectations, or an overwhelming workload, pressure at work can be a major source of stress
2. Money problems: Financial issues can be a significant source of stress, from not being able to make ends meet to struggling with debt.
3. Unhealthy lifestyle: Poor diet, lack of exercise, and sleep deprivation can all contribute to stress.
4. Unhappiness at home: Problems in relationships, conflicts with family members, and other domestic issues can be a major source of stress.
5. Uncertainty: Excessive worry and anxiety about the future, health issues, or other unknowns can be a major cause of stress

Significantly more detailed, nicer/clearer list like formatting

[Dubois et al 2023]

Limitations of RL + Reward Modeling

- Human preferences are unreliable!
  - ”Reward hacking” is a common problem in RL
  - Chatbots are rewarded to produce responses that *seem* authoritative and helpful, *regardless of truth*
  - This can result in making up facts + hallucinations

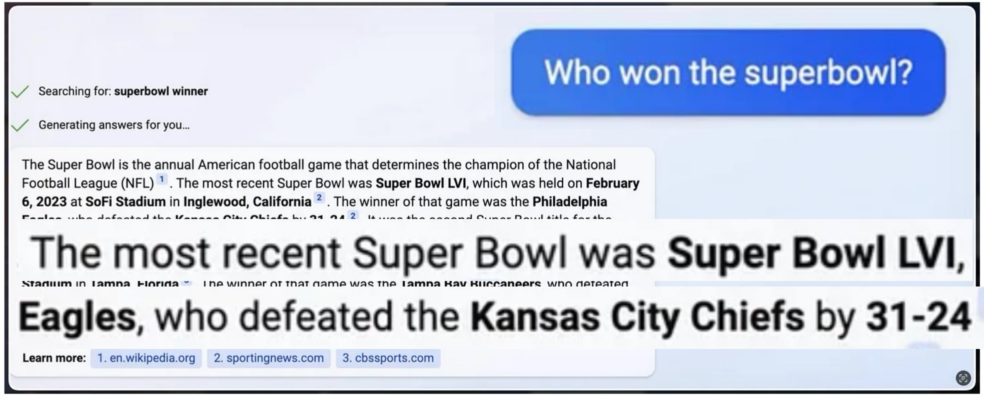
TECHNOLOGY

Google shares drop $100 billion after its new AI chatbot makes a mistake

February 9, 2023 · 10:15 AM ET

https://www.npr.org/2023/02/09/1155650909/google-chatbot--error-bard-shares

Bing AI hallucinates the Super Bowl

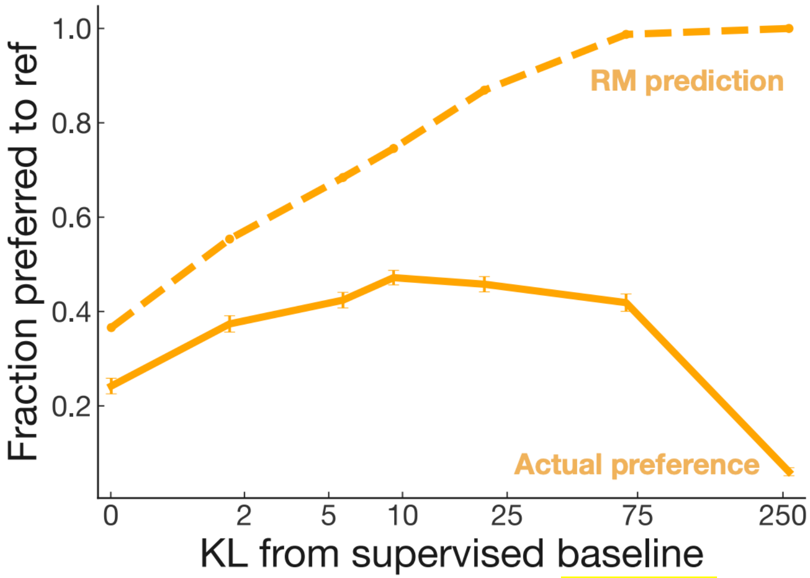


https://news.ycombinator.com/item?id=34776508
https://apnews.com/article/kansas-city-chiefs-philadelphia-eagles-technology-science-82bc20f207e3e4cf81abc6a5d9e6b23a [Slide from CS224n]

Limitations of RL + Reward Modeling

- Human preferences are unreliable!
  - ”Reward hacking” is a common problem in RL
  - Chatbots are rewarded to produce responses that *seem* authoritative and helpful, *regardless of truth*
  - This can result in making up facts + hallucinations
- **Models** of human preferences are *even more* unreliable!

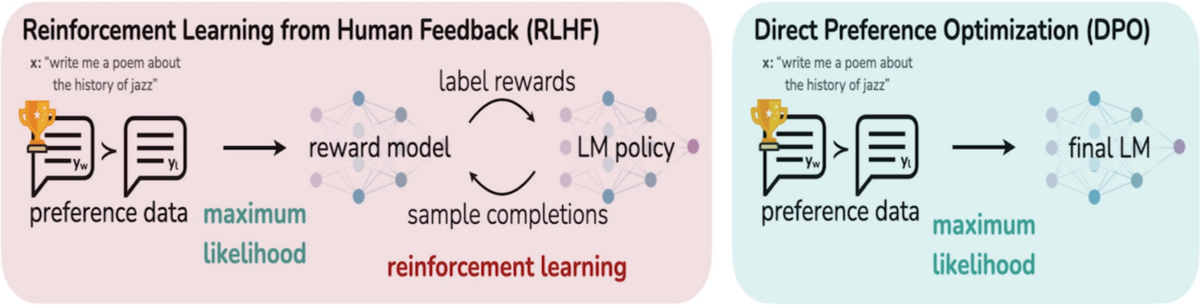
Reward model over-optimization



$$R(s) = RM_{\phi}(s) - \beta \log \left(\frac{p_{\theta}^{RL}(s)}{p^{PT}(s)} \right)$$

[Stiennon et al., 2020]

Removing the ‘RL’ from RLHF



$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$

$$-\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Bigg[\underbrace{\sigma(\hat{r}_\theta(x,y_l) - \hat{r}_\theta(x,y_w))}_{\text{higher weight when reward estimate is wrong}}\Bigg[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l}\Bigg]\Bigg],$$

$$\hat{r}_\theta(x,y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$$

You can replace the complex RL part with a very simple weighted MLE objective
Other variants (KTO, IPO) now emerging too

[Rafailov+ 2023]

What's next?

RLHF is still a very underexplored and fast-moving area!

RLHF gets you further than instruction finetuning, but is (still!) data expensive.

Recent work aims to alleviate such data requirements:

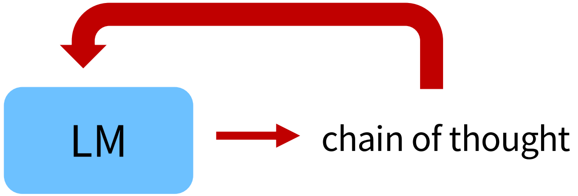
- RL from **AI feedback** [Bai et al., 2022]
- Finetuning LMs on their own outputs [Huang et al., 2022; Zelikman et al., 2022]

However, there are still many limitations of large LMs (size, hallucination) that may not be solvable with RLHF!

LARGE LANGUAGE MODELS CAN SELF-IMPROVE

Jiaxin Huang[1*] **Shixiang Shane Gu**[2] **Le Hou**[2†] **Yuexin Wu**[2] **Xuezhi Wang**[2]
Hongkun Yu[2] **Jiawei Han**[1]
[1]University of Illinois at Urbana-Champaign [2]Google
[1]{jiaxinh3, hanj}@illinois.edu [2]{shanegu, lehou, crickwu, xuezhiw, hongkuny}@google.com

[Huang et al., 2022]



Self-Taught Reasoner (STaR)
[Zelikman et al., 2022]

[Slide from CS224n]

Topics in LLMs + Statistical machine learning

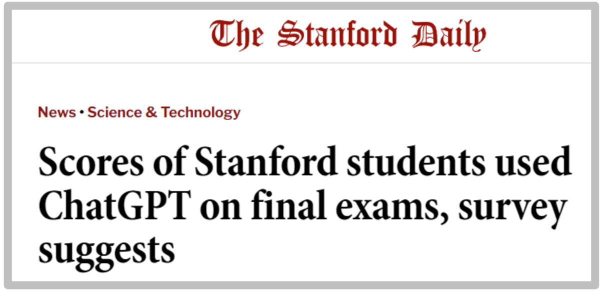
- Conformal inference + Language modeling
- Testing for benchmark contamination
- Watermarking language models

Language models work pretty well.. But.

Air Canada ordered to pay customer who was misled by airline's chatbot

Company claimed its chatbot 'was responsible for its own actions' when giving wrong information about bereavement fare

Hallucinations

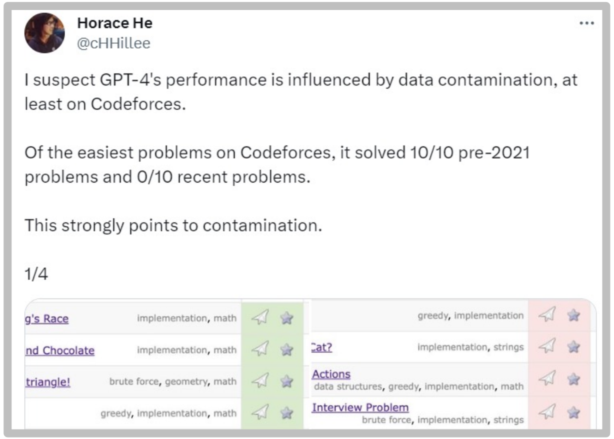


The Stanford Daily

News • Science & Technology

Scores of Stanford students used ChatGPT on final exams, survey suggests

Misuse and spam



Horace He
@cHHillee

I suspect GPT-4's performance is influenced by data contamination, at least on Codeforces.

Of the easiest problems on Codeforces, it solved 10/10 pre-2021 problems and 0/10 recent problems.

This strongly points to contamination.

1/4

g's Race	implementation, math		greedy, implementation
nd Chocolate	implementation, math	Cat?	implementation, strings
triangle!	brute force, geometry, math	Actions	data structures, greedy, implementation, math
	greedy, implementation, math	Interview Problem	brute force, implementation, strings

Untrustworthy evaluations

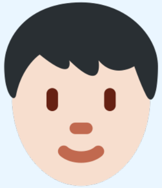
The big picture: we need better tools to study LLMs

We need more than heuristics for LMs

we want precise, controllable statistical guarantees.

Can we give guarantees on correctness?

We can never ensure 100% correctness, but what about *high-probability guarantees?*



Who is Michael Jordan?

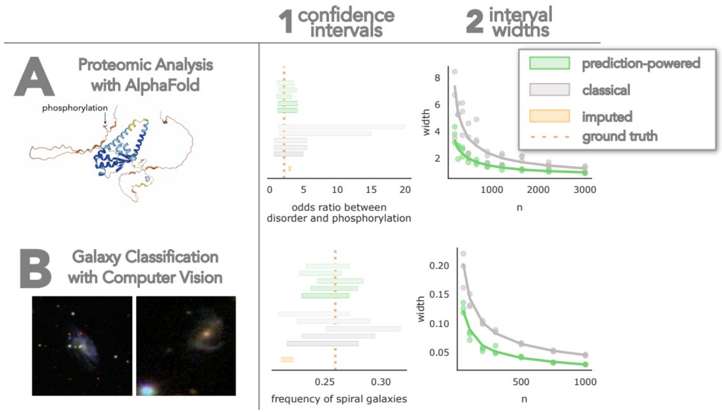
With 95% probability, the following statement is correct:

$F_1(x)$ = `Michael Jordan is a former professional basketball player.`

How can we do this? Conformal prediction

Detour: conformal prediction is a powerful way to get (uncertainty) from black-box models

Can be used to provide tight confidence intervals when predicting proteins, galaxies with DNNs



What does conformal prediction have to do with LMs?

There's a correspondence between *correctness* and *uncertainty quantification*

Q: Where was Abe Lincoln Born?
A: Sinking Spring Farm, Hodgenville, Kentucky

LM outputs

y_0 On a farm in Frankfort, Kentucky

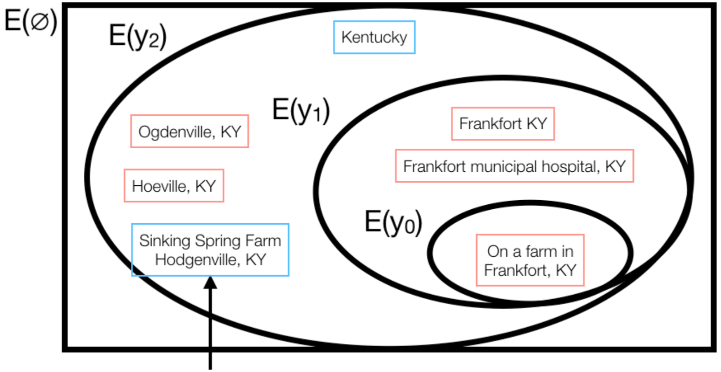
y_1 Frankfort, Kentucky

y_2 Kentucky

(Red shows falsehoods, blue is correct)

Confidence set

$$E(y') := \{y : y \vdash y'\}$$



y*: Sinking Spring Farm, Hodgenville, Kentucky

Key idea: use conformal prediction to "back off"

The algorithm:

1. Construct a sequence of "less specific" y_t
2. Score each y_t via a confidence measure $S(y_t)$
3. Select a cutoff τ for $S(y_t)$ using conformal prediction such that

$$P\left(y^* \in E(y_\tau)\right) \geq 1 - \alpha$$

At inference time return y_τ -- this has a $1 - \alpha$ correctness guarantee

$\mathsf{F}_0(x)$ = `Michael Jordan (born February 17, 1063), is a former professional basketball player.`

$\mathsf{F}_1(x)$ = `Michael Jordan is a former professional basketball player.`

$\mathsf{F}_2(x) = \varnothing$.

Does it work?

One of our results: precise, finite-sample guarantees of factuality / correctness

Theorem 4.1. *Let* $\{X_i, Y_i^*\}_{i=1}^{n+1}$ *be exchangeable,* F_t *be sound, and* $\hat{q}_\alpha$ *be defined as the* $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$*th quantile of the scores* $\{r(X_i, Y_i^*)\}_{i=1}^{n}$*, which we assume to be distinct without loss of generality. Then, for* $\alpha \in \left[\frac{1}{n+1}, 1\right]$*, the following lower bound holds:*

$$\mathbb{P}(Y_{n+1}^* \in \mathsf{E}(\mathsf{F}_{\hat{q}_\alpha}(X_{n+1}))) \geq 1 - \alpha.$$

If $\mathsf{E}(\mathsf{F}_t(\cdot))$ *follows the nested property, then the following upper bound holds:*

$$1 - \alpha + \frac{1}{n+1} \geq \mathbb{P}(Y_{n+1}^* \in \mathsf{E}(\mathsf{F}_{\hat{q}_\alpha}(X_{n+1}))).$$

Soundness: F_t is the empty set for some t

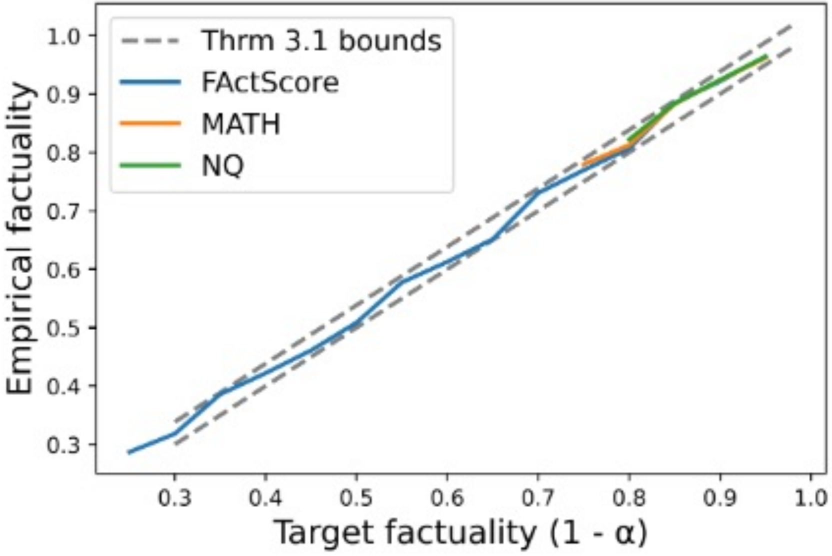
Does it work?

Theorem 4.1. *Let $\{X_i, Y_i^*\}_{i=1}^{n+1}$ be exchangeable, F_t be sound, and $\hat{q}_\alpha$ be defined as the $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$ th quantile of the scores $\{r(X_i, Y_i^*)\}_{i=1}^{n}$, which we assume to be distinct without loss of generality. Then, for $\alpha \in [\frac{1}{n+1}, 1]$, the following lower bound holds:*

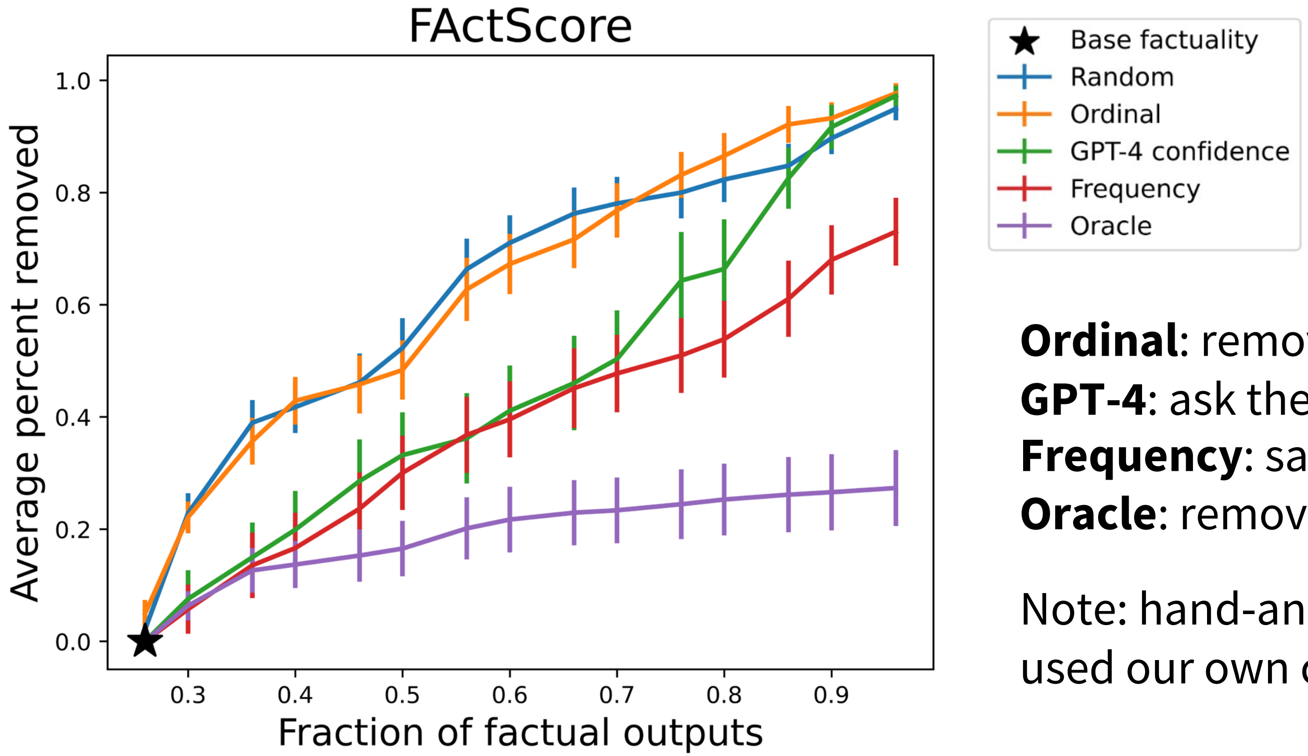
$$\mathbb{P}(Y_{n+1}^* \in \mathsf{E}(\mathsf{F}_{\hat{q}_\alpha}(X_{n+1}))) \geq 1 - \alpha.$$

If $\mathsf{E}(\mathsf{F}_t(\cdot))$ follows the nested property, then the following upper bound holds:

$$1 - \alpha + \frac{1}{n+1} \geq \mathbb{P}(Y_{n+1}^* \in \mathsf{E}(\mathsf{F}_{\hat{q}_\alpha}(X_{n+1}))).$$



Factuality on FactScore from 0% to 80%



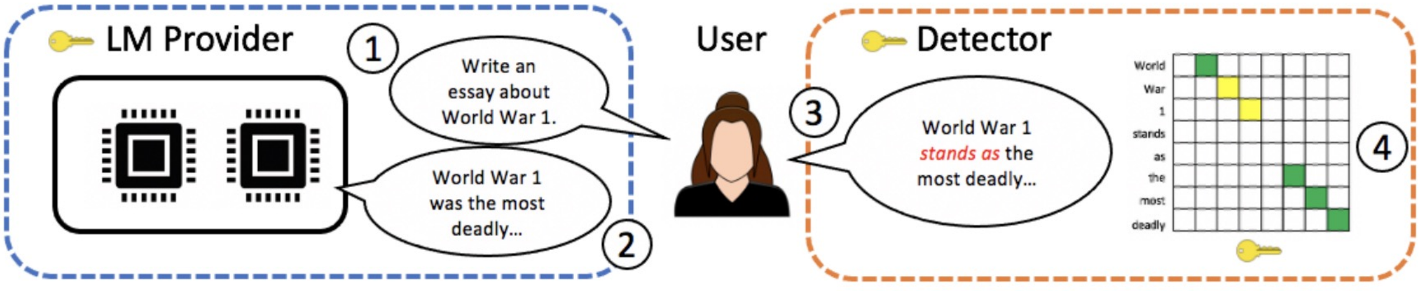
Ordinal: remove the last claims
GPT-4: ask the LM for confidence
Frequency: sample + count claim
Oracle: remove non-factual first

Note: hand-annotated first 50 sam
used our own claim splits for data

Random FactScore examples

Original	Conformal Factuality
Luis Guillermo Rivera is a professional athlete from Colombia who specializes in high jump. He was born on March 20, 1989 in Apartado, Colombia. Rivera has represented Colombia in several international competitions, including the World Championships in Athletics and the Pan American Games. He won the gold medal in the high jump at the 2013 South American Championships in Athletics. His personal best in the high jump is 2.31 meters, which he achieved in 2013.	I'm uncertain about the specific details, and I recommend using a search engine like Google for more accurate and up-to-date information.
Roxana Diaz is a Venezuelan athlete who specializes in the 400 meters hurdles. She represented her country in various international competitions, including the World Championships and the Pan American Games. Diaz is known for her speed, agility, and determination on the track. She has won several medals throughout her career, making her one of the most successful athletes in her country.	Roxana Diaz is an athlete who has represented Venezuela in various international competitions. She has also participated in the World Championships.
Sharad Kumar is an Indian para-athlete who competes in high jump events. He was born on March 1, 1992, in Bihar, India. Kumar was diagnosed with polio at a young age, which affected his left leg. Despite this, he pursued his passion for sports and began training as a high jumper.\n\nKumar has represented India in several international competitions and has won numerous medals. He won a gold medal at the 2014 Asian Para Games in Incheon, South Korea, and a silver medal at the 2018 Asian Para Games in Jakarta, Indonesia. He also won a bronze medal at the 2017 World Para Athletics Championships in London.\nnIn 2021, Kumar won a bronze medal in the men's high jump T42 category at the Tokyo Paralympics, marking his first Paralympic medal. His personal best in high jump is 1.90 meters, which he achieved at the 2017 World Para Athletics Championships.	Sharad Kumar is an Indian para-athlete who was born on March 1, 1992, in Bihar, India. He competes in high jump events and despite his condition, he pursued his passion for sports and began training as a high jumper. Sharad Kumar has represented India in several international competitions and has won numerous medals in his career. Most notably, in 2021, he won a bronze medal in the men's high jump T42 category at the Tokyo Paralympics.
Katsunosuke Hori is a Japanese actor and voice actor. He was born on November 28, 1941 in Tokyo, Japan. He is best known for his work in anime and video games, including roles in "Neon Genesis Evangelion", "Ghost in the Shell: Stand Alone Complex", and "Final Fantasy VII: Advent Children". He is also known for his dubbing roles in foreign films, such as the voice of Gandalf in the Japanese version of "The Lord of the Rings" trilogy.	Katsunosuke Hori is a Japanese actor and voice actor. He is known for his work in anime and video games.
Ra Jong-yil is a South Korean diplomat and academic. He served as the South Korean ambassador to the United Kingdom and Japan. He also served as the deputy director of the National Intelligence Service. As an academic, he has taught at various universities and written several books on North Korea and international relations.	Ra Jong-yil is a South Korean diplomat and academic who has served as the South Korean ambassador to both the United Kingdom and Japan. In addition to his diplomatic work, he has written several books on North Korea and international relations.

Provenance via watermarking

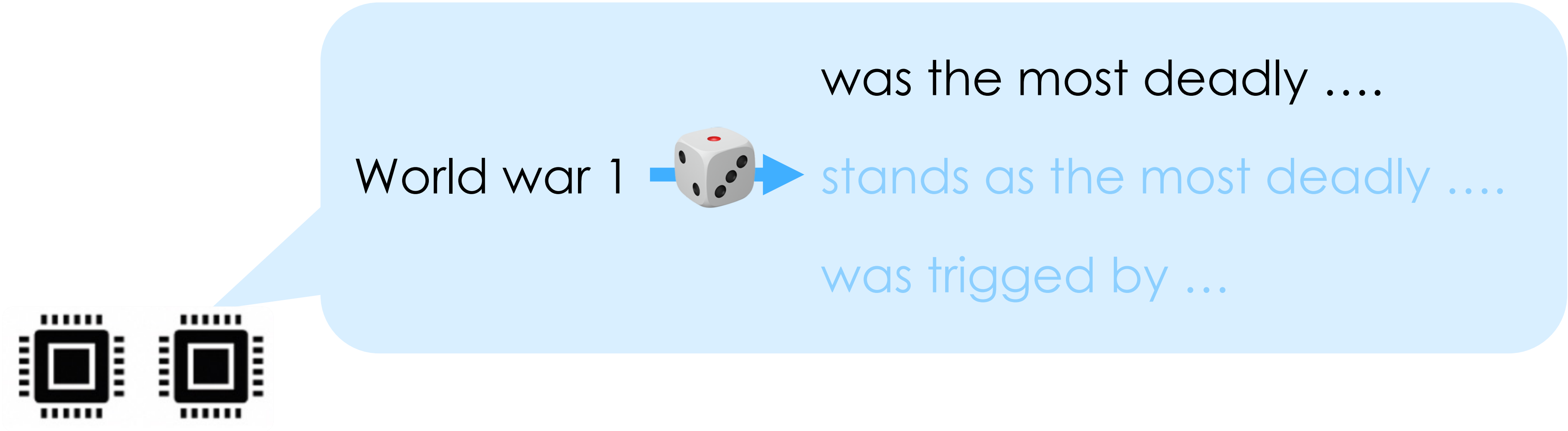


Watermarking enables tracking of LLM-generated text (Kirchenbauer et al)

Challenges:

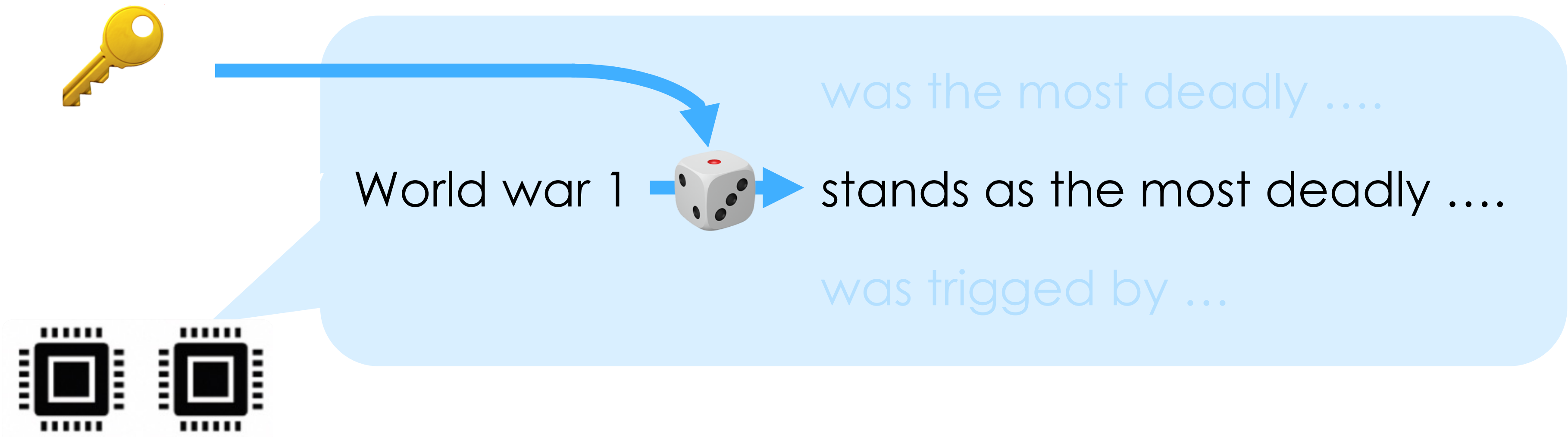
- Watermarks induce distortion (hard sell for LLM vendors)
- Many watermarks highly non-robust (to deletion of a few words, or cropping)

How watermarking works (I)



LMs generate text stochastically

How watermarking works (II)



Watermarks manipulate the randomness
to make outputs detectable

Development of a distortion-free, robust watermark.

In recent work [Kuditipudi et al 2023], we derive a distortion free and robust watermark.

Generate (for each token y_i)

- Draw a random sequence $\xi_i \in [0,1]$, call this the key
- Sample according to $\min_i -\log \xi_i / p_i$ (From Aaronson)

This is distortion free (i.e. the marginal distribution over ξ is p)

Detect

- Find the min-Levenshtein cost with $d(y, \xi) = \sum_i \log(1 - \xi_{i,y_i})$
- Compare vs the min-Levenshtein cost w/ random ξ

This is robust (i.e. can detect under small Levenshtein edits)

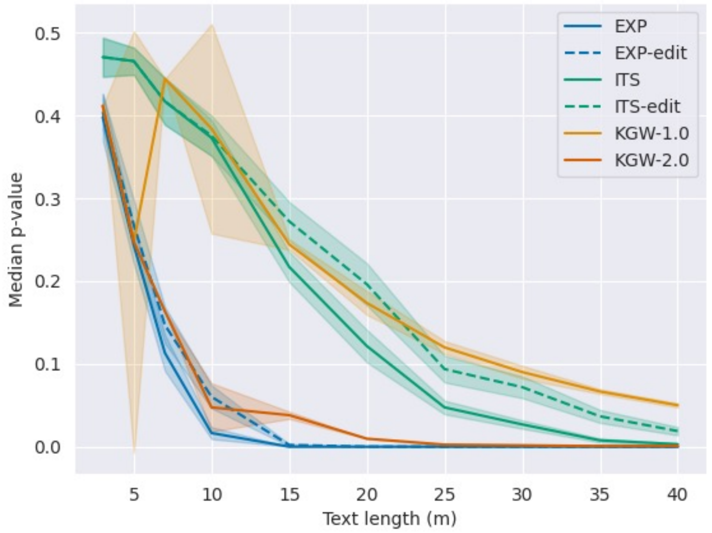
[Kuditipudi et al 2023]

Many interesting watermarks for language models

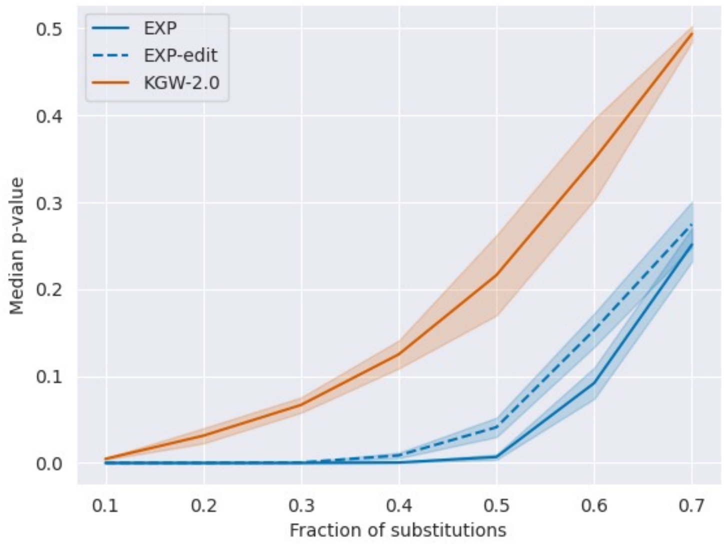
- **Kirchenbauer et al.** bias next token distribution towards a pseudorandom "green list"
- **Aaronson** samples next token as a pseudorandom function of previous k tokens
- **Christ et al.** varying k depending on token probabilities to reduce hash collisions – smart choice of k makes outputs effectively distortion free

Watermark detectability

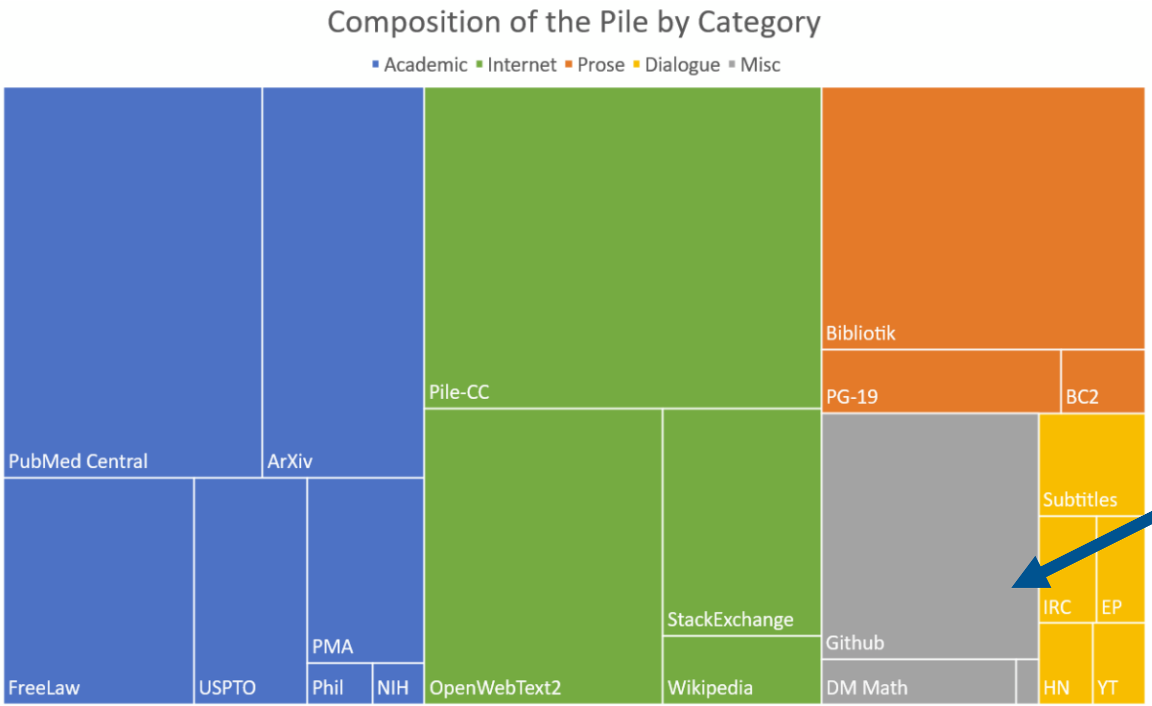
Varying text length



Robust to random deletions



What is in the training data of a LLM?



.. But maybe your test set is in here?

CODEFORCES
Sponsored by TON

Language models derive their strength from massive, lightly-curated pretraining data

We need third party, provable audits of contamination

Horace He
@cHHillee

I suspect GPT-4's performance is influenced by data contamination, at least on Codeforces.

Of the easiest problems on Codeforces, it solved 10/10 pre-2021 problems and 0/10 recent problems.

This strongly points to contamination.

1/4

Susan Zhang
@suchenzang

I think Phi-1.5 trained on the benchmarks. Particularly, GSM8K.

Susan Zhang @suchenzang · Sep 12
Let's take github.com/openai/grade-s...

If you truncate and feed this question into Phi-1.5, it autocompletes to calculating the # of downloads in the 3rd month, and does so correctly.

Change the number a bit, and it answers correctly as well.

1/🕵

Many public claims (and heuristic tests) of contamination – but no proof or audits

The goal: provably detecting test set contamination

Goal: provide a provable (false positive) guarantee for detecting test set contamination.

Our setup:

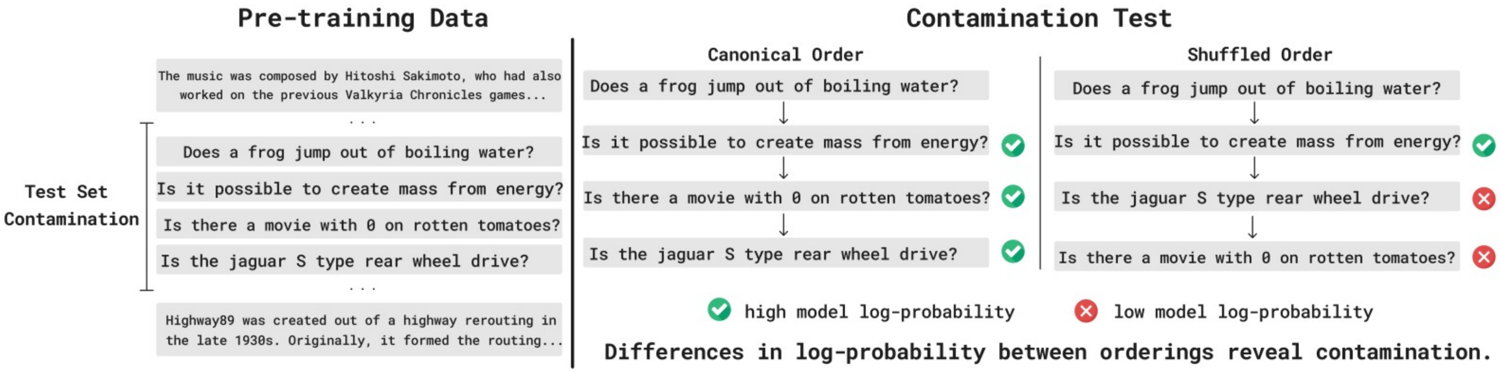
Given a test set and access to log-probabilities from a language model

Return a statistical test for contamination with type-I error rate at most α

1. The null hypothesis is that the test set and model are independent r.v.s
2. The error guarantee should hold w.r.t draws of the datasets

Our approach: exploit the exchangeability of datasets

Our starting observation: most test sets are *exchangeable.*



Key idea:

Language model preference for 'canonical' orderings *must* come from contamination

Our approach: exploit the exchangeability of datasets

Key idea:

Language model preference for 'canonical' orderings *must* come from contamination

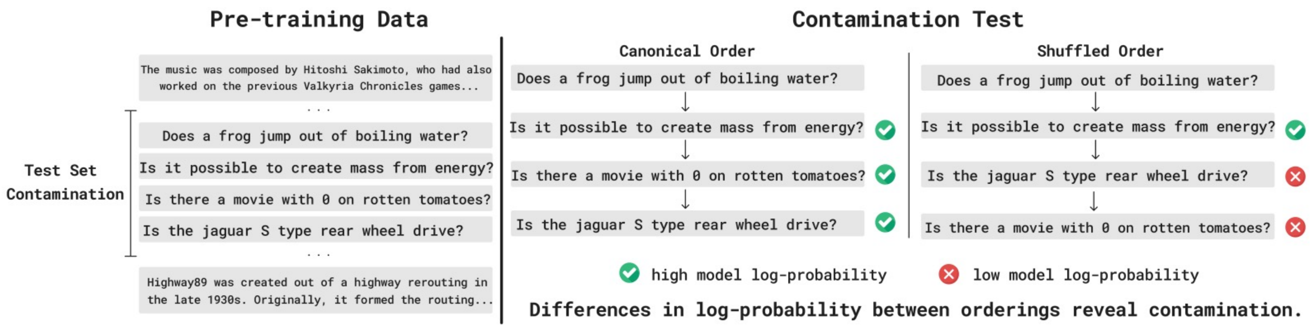
Proposition 1. *Let $seq(X)$ be a function that takes a dataset X and concatenates the examples to produce a sequence, and let X_π be a random permutation of the examples of X where π is drawn uniformly from the permutation group. For an exchangeable dataset X and under H_0,*

$$\log p_\theta(seq(X)) \overset{d}{=} \log p_\theta(seq(X_\pi)).$$

Proof This follows directly from the definitions of exchangability and H_0. Since X is exchangable, $\text{seq}(X) \overset{d}{=} \text{seq}(X_\pi)$ and by the independence of θ from X under H_0, we know that $(\theta, \text{seq}(X)) \overset{d}{=} (\theta, \text{seq}(X_\pi))$. Thus, the pushforward under $\log p_\theta(\text{seq}(X))$ must have the same invariance property. $\square$

This leads to a simple test for contamination

Shuffle and compute log probs



Gives exact p-values

$$\hat{p} := \frac{\sum_{i=1}^{m} \mathbb{1}\{\log p_\theta(\mathrm{seq}(X)) < \log p_\theta(\mathrm{seq}(X_{\pi_m}))\} + 1}{m+1}.$$

Result #1 – detection in known settings

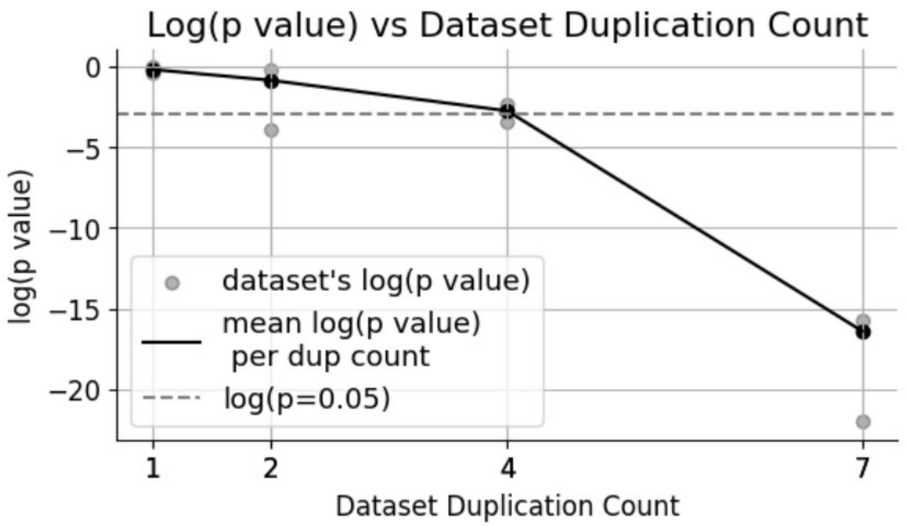
Can we detect known contamination?

We pretrained a 1.4B param, 20B token LM w/ known contamination

Name	Size	Dup Count	Permutation p	Sharded p
BoolQ	1000	1	0.099	0.156
HellaSwag	1000	1	0.485	0.478
OpenbookQA	500	1	0.544	0.462
MNLI	1000	10	**0.009**	**1.96e-11**
Natural Questions	1000	10	**0.009**	**1e-38**
TruthfulQA	1000	10	**0.009**	**3.43e-13**
PIQA	1000	50	**0.009**	**1e-38**
MMLU Pro. Psychology	611	50	**0.009**	**1e-38**
MMLU Pro. Law	1533	50	**0.009**	**1e-38**
MMLU H.S. Psychology	544	100	**0.009**	**1e-38**

100% detection rate on ≥ 10 duplication count datasets

Result #2 – detection even in low duplication count



Around 50% detection rate between 2 to 4 duplicates

Result #3 – contamination in the wild

Dataset	Size	LLaMA2-7B	Mistral-7B	Pythia-1.4B	GPT-2 XL	BioMedLM
AI2-ARC	2376	0.318	**0.001**	0.686	0.929	0.795
BoolQ	3270	0.421	0.543	0.861	0.903	0.946
GSM8K	1319	0.594	0.507	0.619	0.770	0.975
LAMBADA	5000	0.284	0.944	0.969	0.084	0.427
NaturalQA	1769	0.912	0.700	0.948	0.463	0.595
OpenBookQA	500	0.513	0.638	0.364	0.902	0.236
PIQA	3084	0.877	0.966	0.956	0.959	0.619
MMLU†	–	0.014	0.011	0.362	–	–

- No evidence of contamination (except ARC+Mistral)
- MMLU tests consistent with Touvron et al's contamination test

Key ideas

- **Pretraining**: scaling laws enable targeted scaling

- **Post-training:** RLHF and instruction tuning lead to systems like chatGPT that can follow user-instructions.

- **Statsml + LLMs:** Many interesting new problems that can be addressed with precise, statistical tools