Pretraining II: Decoding, Scaling, Prompting, and Beyond

CSE 5525: Foundations of Speech and Natural Language Processing

https://shocheen.github.io/courses/cse-5525-spring-2026

THE OHIO STATE UNIVERSITY

Logistics

- Final Project Proposal: Due tonight.

- Homework 3 will be released tonight.
  - Topic: prompting/finetuning models for code generation (text-to-SQL)

Last Class Recap: Masked LMs

- Denoising pretraining objectives
  - BERT
  - T5
  - BART

- Decoding algorithms

Going forward – decoder only LMs

- We will (mostly) talk about decoder only LMs going forward as they are most commonly used now
    - Like GPTs, Claude, Llama models, Mistral, Qwen, and many more.

Why decoder only LMs?

Next-word Prediction

- Language models do next word prediction

- At first look, next-word-completion seems like a very simple task

- Why does it make sense to focus on it so much?

Why decoder only LMs?
Why Does it Make Sense?

The woman walked across the street, checking for traffic over ___

Why decoder only LMs?
Why Does it Make Sense?

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

Why decoder only LMs?
Why Does it Make Sense?

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

Why decoder only LMs?
Why Does it Make Sense?

Iroh went into the kitchen to make some tea. Standing next to Iroh, Zuko pondered his destiny. Zuko left the ____

Why decoder only LMs?
Why Does it Make Sense?

I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21, _____

Why decoder only LMs?
Why Does it Make Sense?

Ohio State is located in _____

Why decoder only LMs?

Why Does it Make Sense?

- The woman walked across the street, checking for traffic over ___ [coreference]

- I went to the ocean to see the fish, turtles, seals, and ____ [lexical semantics / topics]

- 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]

- I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21, _____ [reasoning]

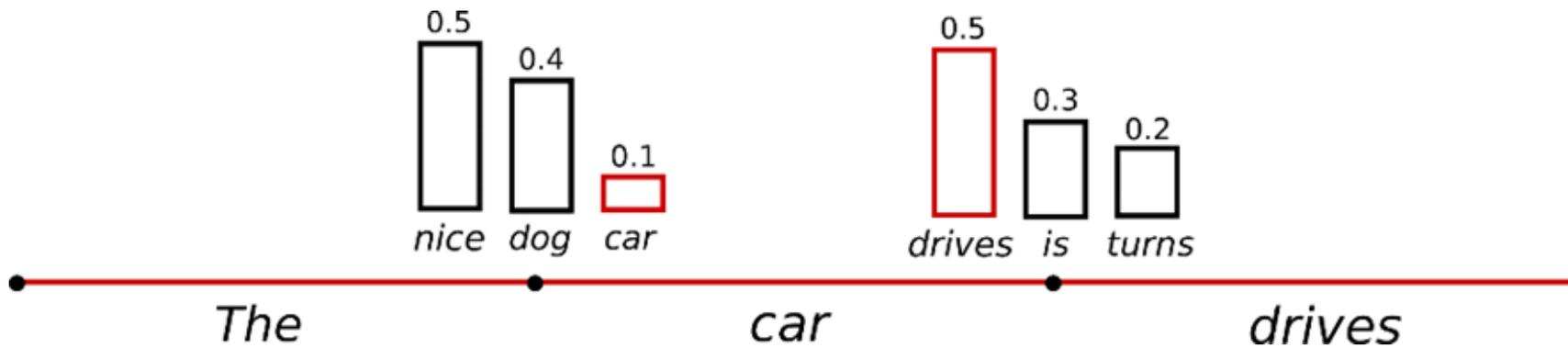- Ohio State is located in _____ [knowledge]

The learned representations have to account for a lot to succeed in this seemingly straightforward task

Decoding strategies

Sampling

- Sampling:

$$x_i \sim p(x_i|x_1, \ldots, x_{i-1}) \text{ until } x_i = \text{STOP}$$

This can often generate incoherent gibberish

Greedy Decoding

- Greedy (i.e., $\arg max$):

$$x_i = \arg\max_{x_1 \in \mathcal{V}} p(x_i | x_1, \ldots, x_{i-1}) \text{ until } x_i = \text{STOP}$$

- How many different strings can we generate this way?

Top-k sampling [Fan et al., 2018]

Filter k most likely next tokens and redistributed the probability mass among only those k tokens, then sample from the new distribution

Problem: It doesn't dynamically adapt the number of words that are filtered from the next word probability distribution

Reasonable candidates (left fig) are eliminated, and ill-fitted (right fig) are not

Source: https://huggingface.co/blog/how-to-generate
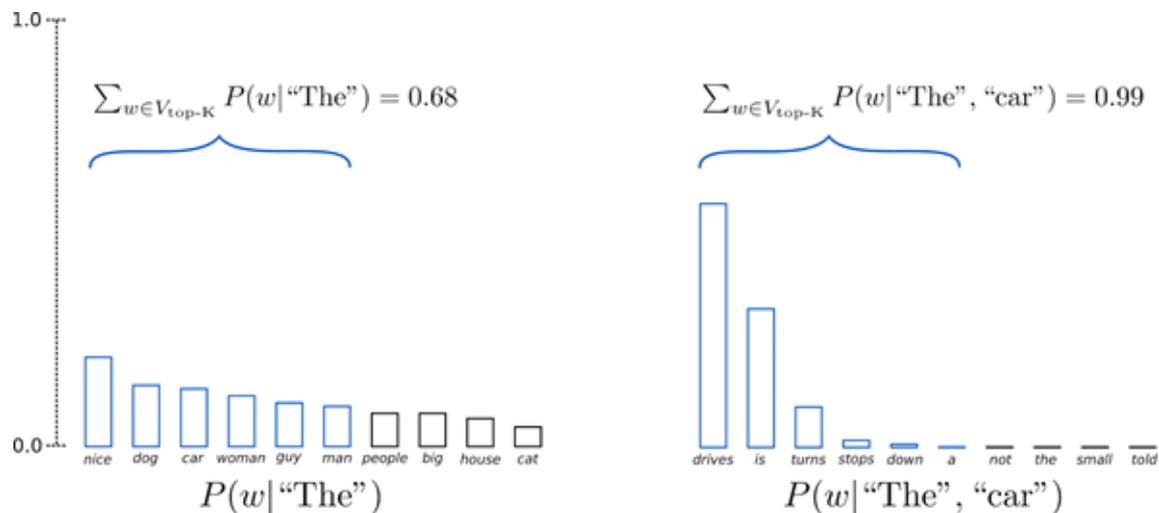
Top-p (nucleus) sampling [Holtzman et al., 2020]

Sample from the smallest possible set of tokens whose cumulative probability exceeds the probability p
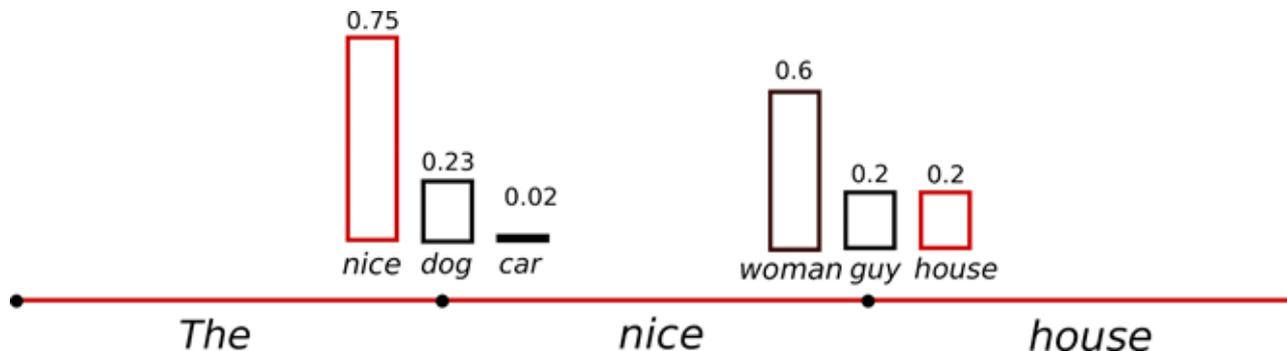
Source: https://huggingface.co/blog/how-to-generate
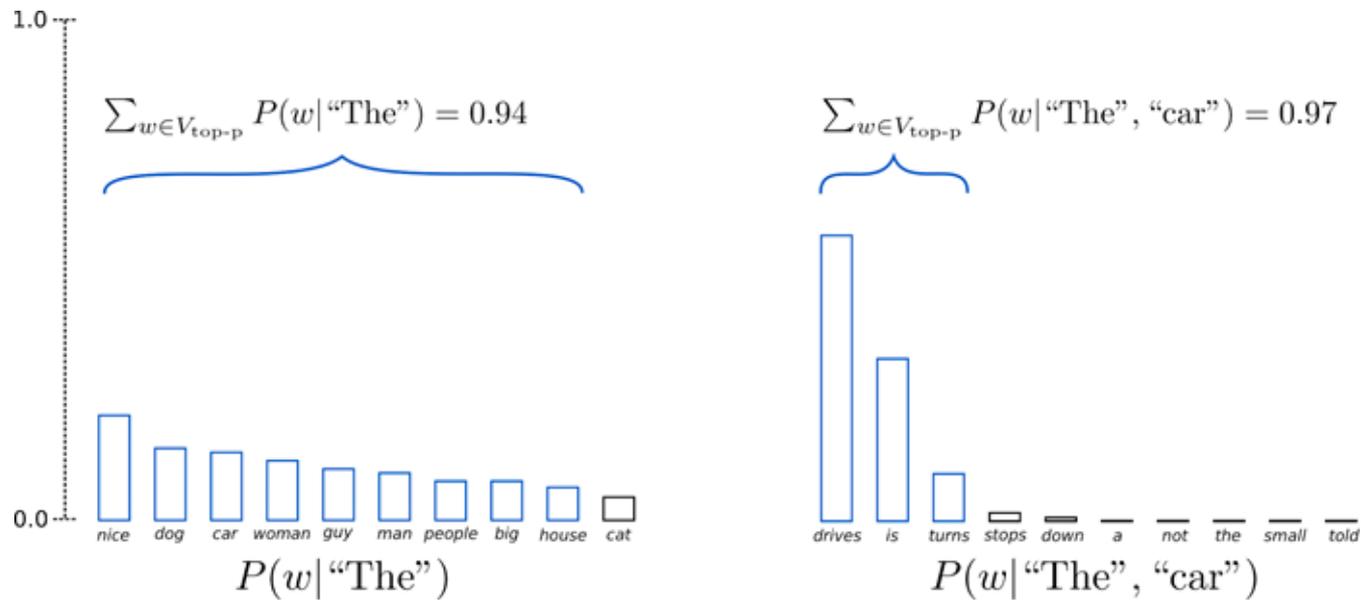
Adjusting Distribution **Temperature**

- Add a temperature parameter to the softmax
  - Given z is the vector with logits, and $T \in \mathbb{R}$ in the temperature
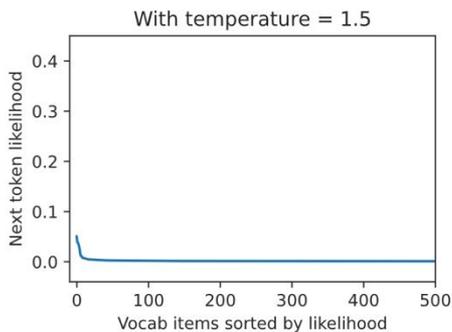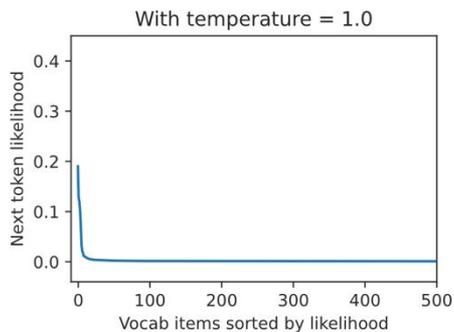
$$p(y_i \mid x) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$
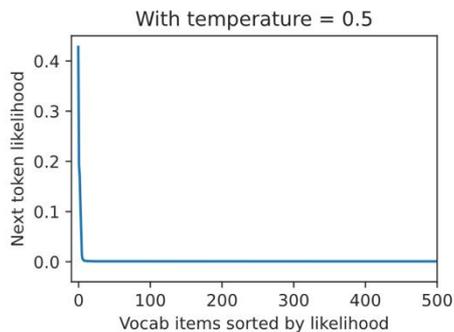
Higher T: softens probabilities.
Lower T: sharpens probabilities.

Adjusting Distribution Temperature

- What happens with $T = 1$? $T = 0$ (or almost)? $T \in [0,1)$? $T > 1$?

Decoding

- Various decoding techniques: greedy, sampling, temperature-based, top-k, nucleus

- Most common: temperature-based + top-p sampling

- Are these optimal decoding strategies?

Decoding

- Various decoding techniques: greedy, sampling, temperature-based, top-k, nucleus

- Most common: temperature-based + top-p sampling

- Are these optimal decoding strategies? As in will it get you the highest probability sequence?

Output 1	0,2	0,5	0,1
	I	love	Lucy

Output 2	0,2	0,1	0,99
	I	hate	Lucy

Greedy decoding/search
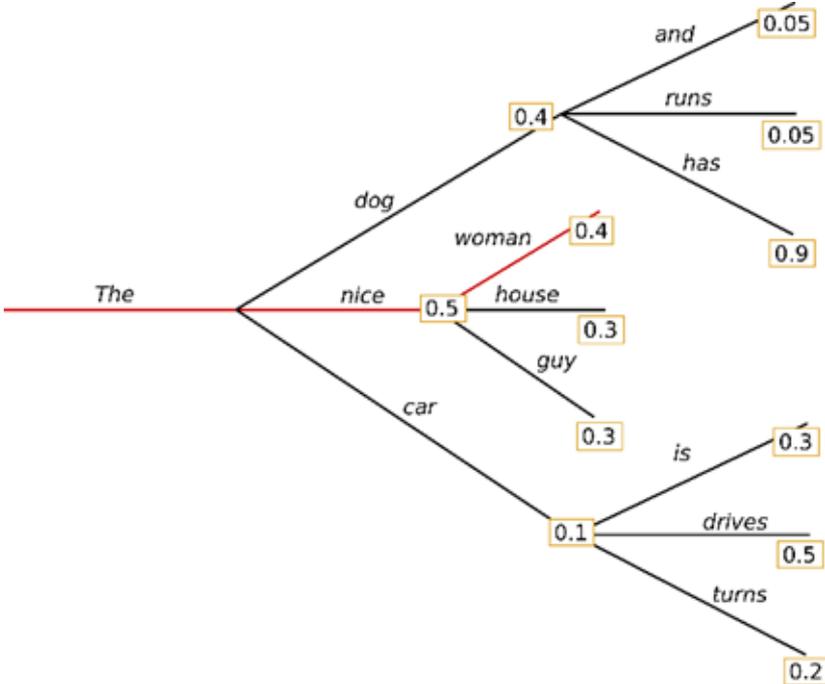
Source: https://huggingface.co/blog/how-to-generate

Beam Search

- Sampling techniques are not optimal

  - Following a single hypothesis is just not sufficient, but enumerating all is intractable

- **Beam search** is middle ground

  - Follow a set of hypothesis, always keeping the top ones

  - The size of the set B is a hyperparameter
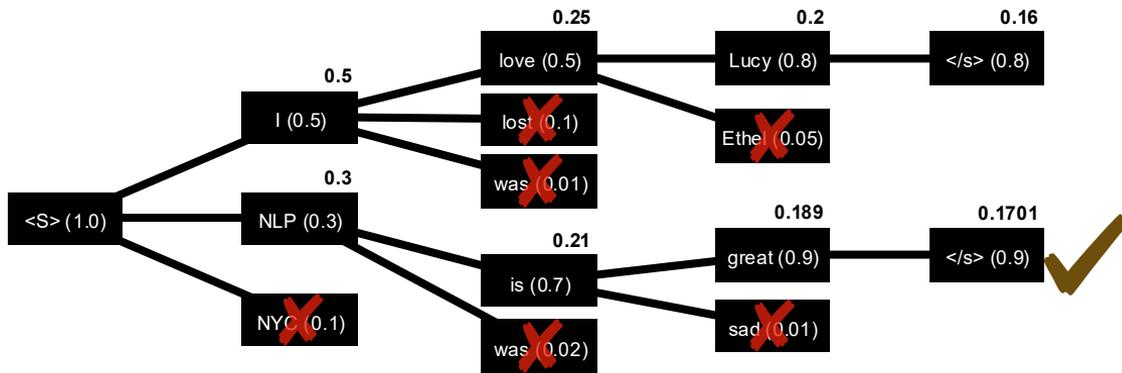
Beam Search

- Sampling techniques are not optimal
  - Following a single hypothesis is just not sufficient, but enumerating all is intractable
- Beam search is middle ground
  - Follow a set of hypothesis, always keeping the top ones
  - The size of the set B is a hyperparameter
  - It's an approximation method
  - What happens with $B = 1$? $B = \infty$?
  - What is the cost of beam search compared to the sampling techniques we saw?
  - Can you combine sampling techniques with beam search?

Scaling

Some History: the GPTs

GPT [[Radford et al. 2018](#)]

- Transformer LM released in 2018 by OpenAI

- Decoder with 12 transformer blocks, 117M parameters, 768-dimensional hidden states, 3072-dimensional feed-forward hidden layers; BPE with 40k merges

- Trained on BookCorpus: over 7,000 unique books: long spans of contiguous text for learning long-distance dependencies

- Impressive results when fine-tuned on several NLP tasks: Entailment, textual similarity, multiple choice questions

Some History: the GPTs

GPT-2 [Radford et al. 2018]

- GPT-2 scaled the models to 1.5B parameters

- Increasingly convincing generations

- Impressive results on benchmarks

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

Some History: the GPTs

GPT-3 [Brown et al. 2020]

- GPT-3 scaled the model size to 175B parameters

- So far, two ways of interaction with models:

  - Sample from the distribution (generation)

  - Fine-tune on a specific task

- GPT-3 demonstrated few-shot learning **without parameter updates — In-context Learning (ICL)**

  - In-context examples seem to specify the task, allowing the model to complete it on a new input
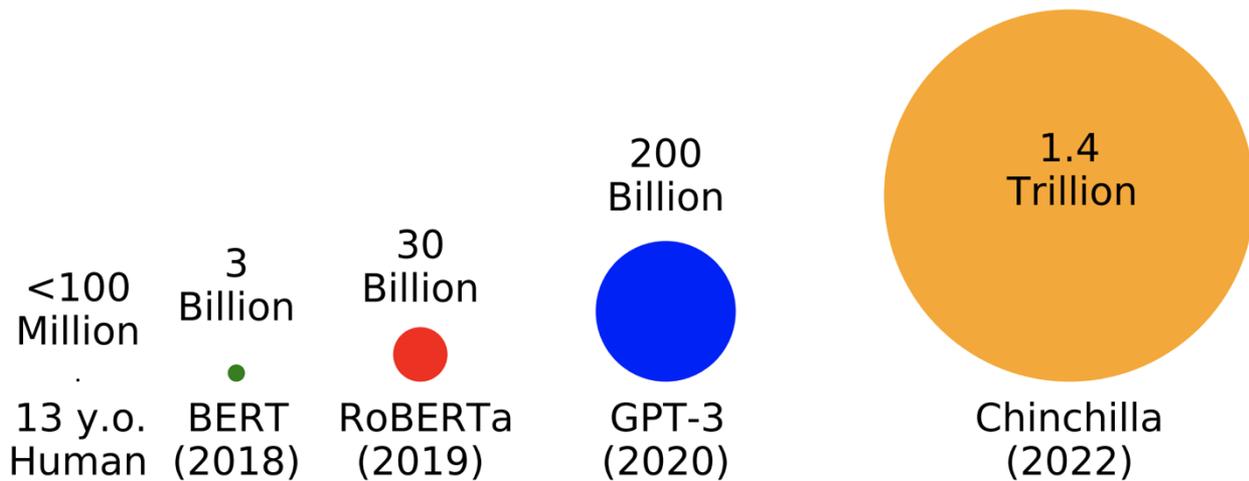
  - More on this later on ...

Scaling Up

- Two dimensions of scaling up:
    - **Data:** the number of raw tokens the learner is given
    - **Parameters:** the number of parameters in the model
- All this requires scaling up **compute**
    - Storage (memory, disk space, etc), GPUs, networking

Scaling Up

Data

- Largest models in the past couple of years hover around max of 500B

<100
Million

.

13 y.o.
Human

3
Billion

BERT
(2018)

30
Billion

RoBERTa
(2019)

200
Billion

GPT-3
(2020)

1.4
Trillion

Chinchilla
(2022)

Scaling Up

Data

- How do we get text data at scale?

- Scrape whatever we can get from the web

  - Seed webcrawler with initial URLs

  - Identify new URLs via outlinks

  - Download HTML pages, extract raw text, postprocess text

- Done? Not really …

  - The Internet is a mess

  - What would you do next?

Data
Web Scraping: Filtering Heuristics

- Deduplication

- Remove "low-quality" data

- Remove non-English data a language classifier

- Remove stuff your model probably is better off without: personally identifiable information, adult content, hate speech, copyrighted data, NLP benchmarks (why?)

Data
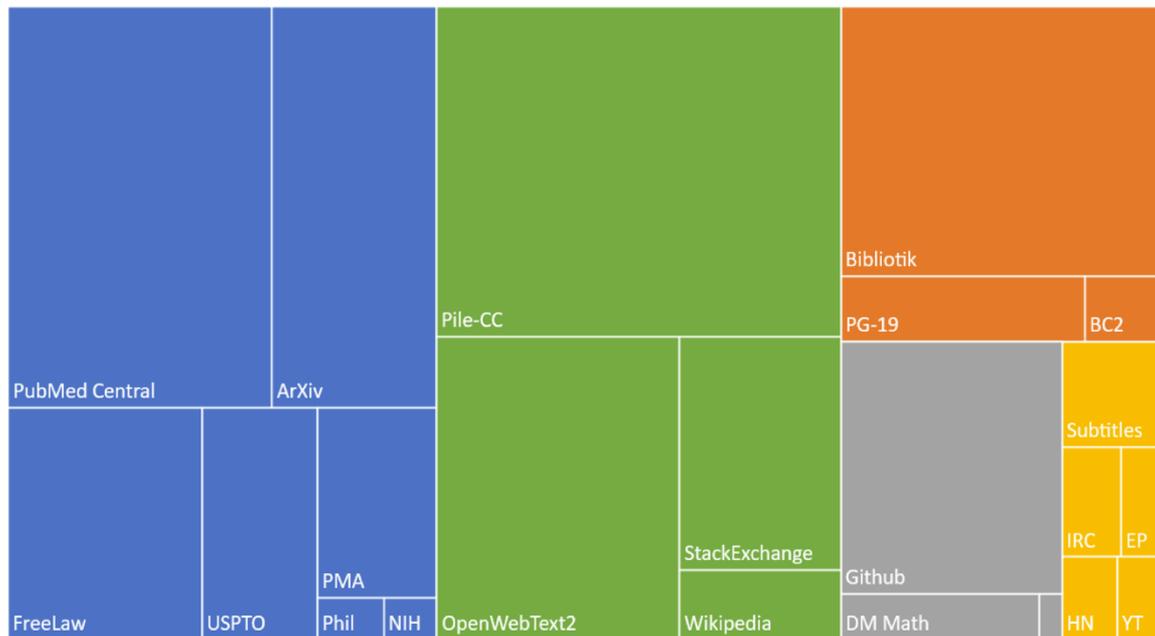## Web Scraping: Filtering Tradeoffs

- Personally identifiable information
  - But what about the phone numbers of public companies?

- Adult content and hate speech
  - Very culturally dependent

- Copyrighted data
  - How to identify? Is it fair use?

Data
Composition: the Pile

Composition of the Pile by Category

■ Academic ■ Internet ■ Prose ■ Dialogue ■ Misc

[Gao et al. 2020]

Data
Large Raw Text Corpora

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]
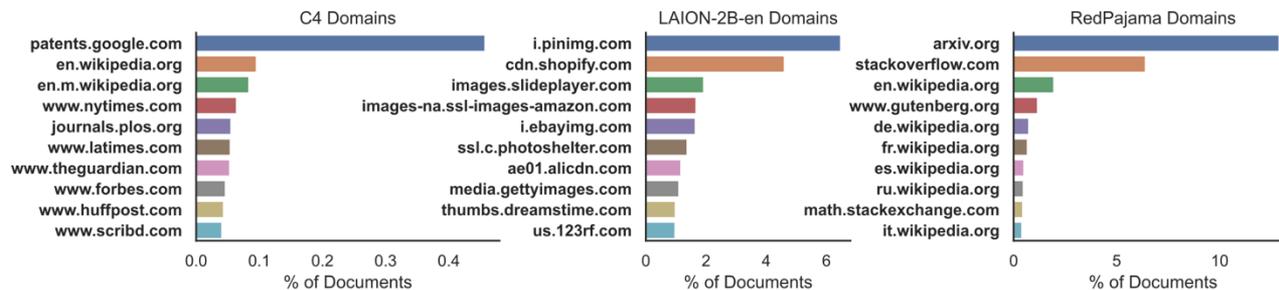
Data

What is the Web Missing?

- Many non-English languages are very limited

- Dialects with fewer speakers.

- Non-written languages (e.g., American Sign Language)

- Language from people not on the web

All this comes to reinforce biases, which impact the technology available to people
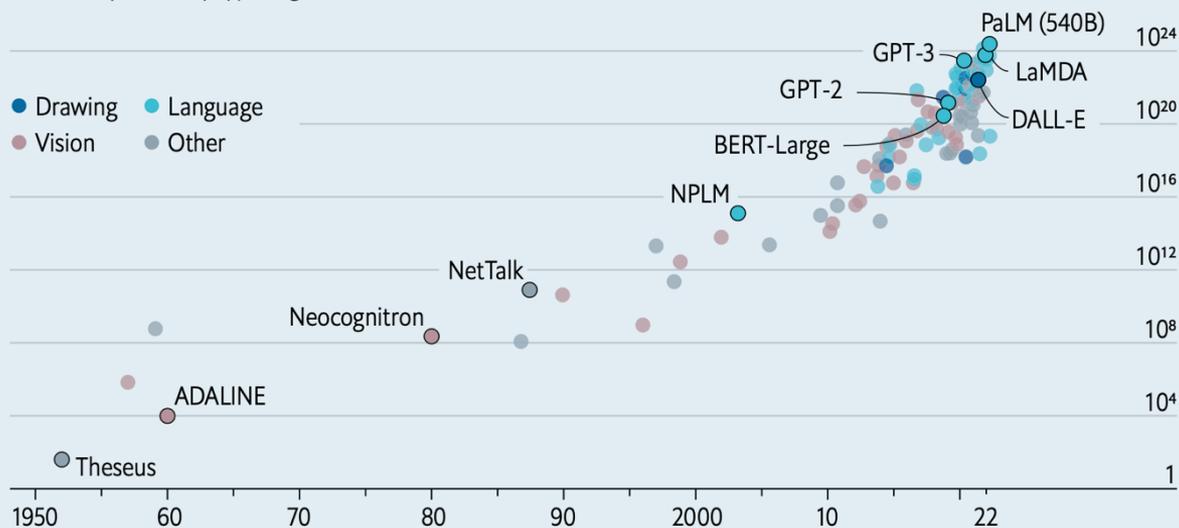
Scaling Up
Compute

The blessings of scale
AI training runs, estimated computing resources used, floating-point operations
Selected systems, by type, log scale

Legend:
- Drawing
- Language
- Vision
- Other

Labeled points: PaLM (540B), GPT-3, LaMDA, GPT-2, DALL-E, BERT-Large, NPLM, NetTalk, Neocognitron, ADALINE, Theseus

Y-axis: 10^{24}, 10^{20}, 10^{16}, 10^{12}, 10^8, 10^4, 1

X-axis: 1950, 60, 70, 80, 90, 2000, 10, 22

Sources: "Compute trends across three eras of machine learning", by J. Sevilla et al., arXiv, 2022; Our World in Data
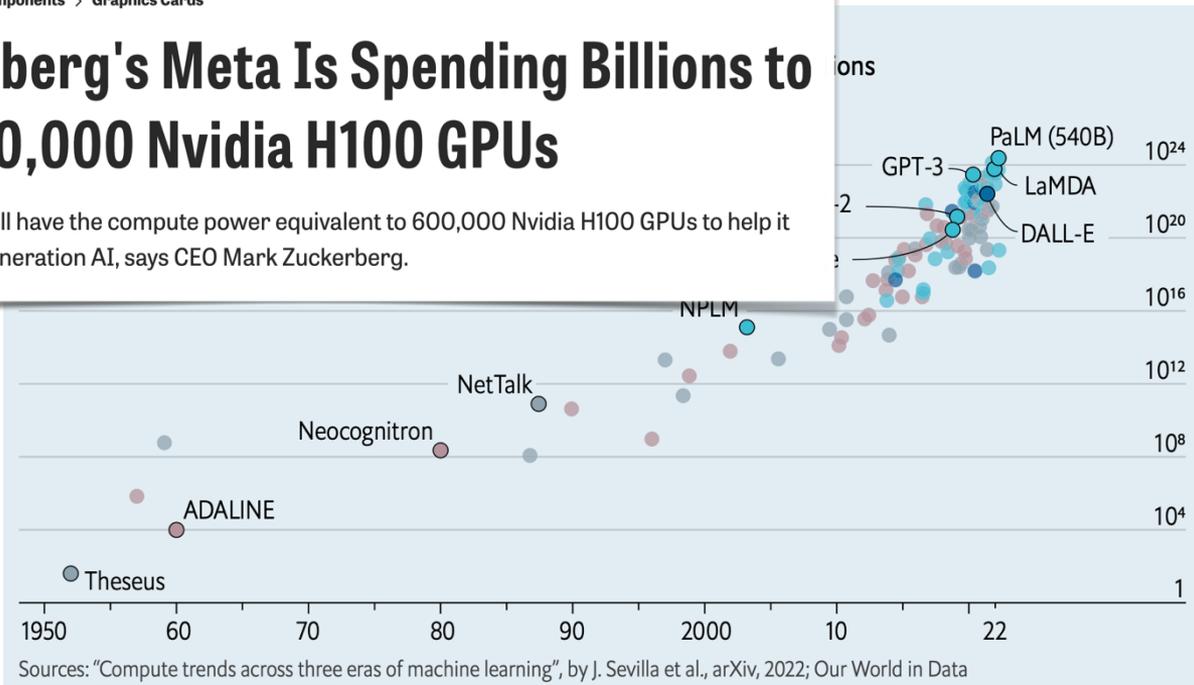
Scaling Up
Compute

Home > News > Components > Graphics Cards

Zuckerberg's Meta Is Spending Billions to Buy 350,000 Nvidia H100 GPUs

In total, Meta will have the compute power equivalent to 600,000 Nvidia H100 GPUs to help it develop next-generation AI, says CEO Mark Zuckerberg.
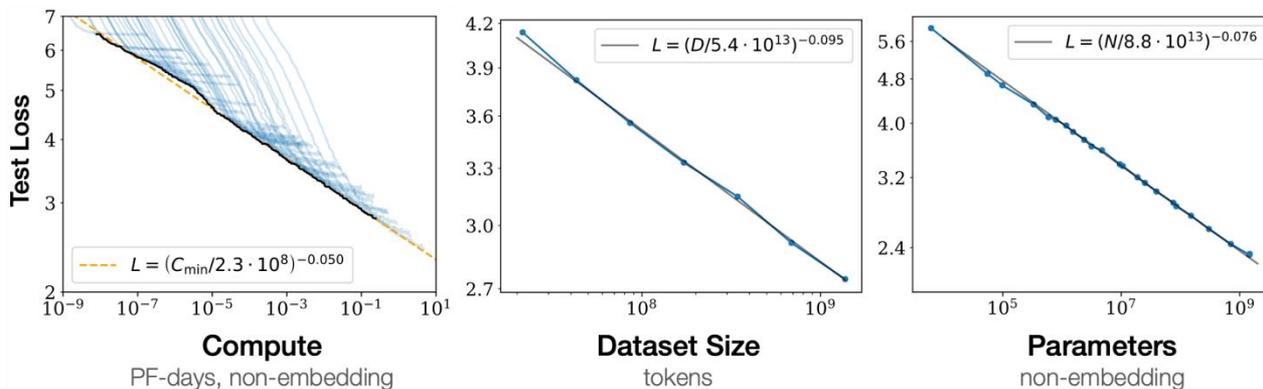
...ions

PaLM (540B) 10^{24}

GPT-3 LaMDA

-2 DALL-E 10^{20}

10^{16}

NPLM

10^{12}

NetTalk

Neocognitron 10^{8}

ADALINE 10^{4}

Theseus 1

1950   60   70   80   90   2000   10   22

Sources: "Compute trends across three eras of machine learning", by J. Sevilla et al., arXiv, 2022; Our World in Data

Scaling Up Impact

How Does Performance Improve?

- When we scale up…
    - The model size
    - The number of training examples
    - The batch size
   - The number of model updates (i.e., training longer)

Scaling Laws

The three plots show Test Loss versus Compute (PF-days, non-embedding) with $L = (C_{min}/2.3 \cdot 10^8)^{-0.050}$, Dataset Size (tokens) with $L = (D/5.4 \cdot 10^{13})^{-0.095}$, and Parameters (non-embedding) with $L = (N/8.8 \cdot 10^{13})^{-0.076}$.

- Empirical test loss has a power law relationship with each individual factor
- Transformers scale well, and in a very predictable way

[Kaplan et al. 2020]

Scaling Laws

- Larger models require fewer samples to reach the same performance
- The optimal model size grows smoothly with the loss target and compute budget
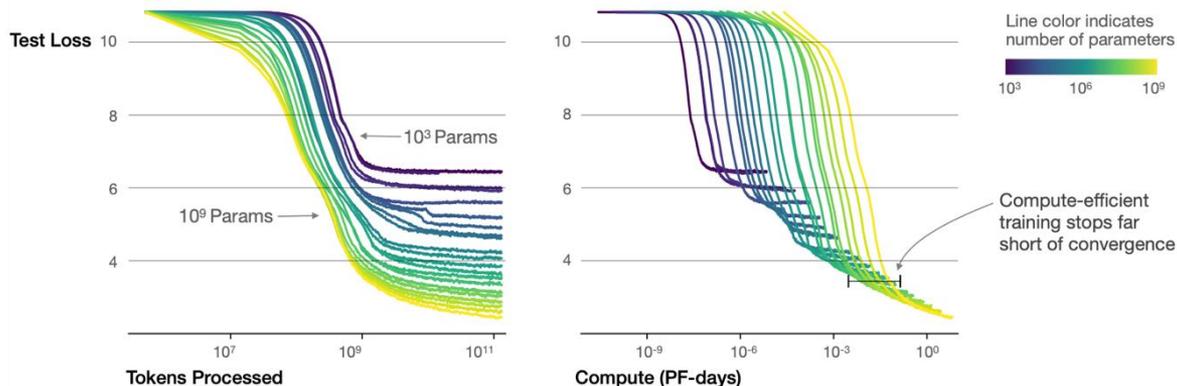
[Kaplan et al. 2020]

Scaling Laws

- Scaling laws allow us to predict the loss:
  - Given a compute budget, how should we scale the data and number of parameters to get the best model?

- Scaling laws were identified by [Kaplan et al. 2020](#), and later refined by [Hoffmann et al. 2022](#)

- The papers also provide exact formulas with coefficients for the Transformer architectures they used

- There have been papers since then refining these laws.

Scaling Up
What Do We Get?

- I put _____ form 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 / topics]
- 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]
- I was thinking about the sequence that goes 1, 1, 2, 3, 5, 8, 13, 21, _____ [reasoning]
- Cornell Tech is located in _____, New York [knowledge]

The learned representations have to account for a lot to succeed in this seemingly straightforward task

- We get really expressive representations

- Very impressive generations

- But how useful are these models?
  - Not that useful, yet
  - But: we can fine-tune them to be very useful
    - This is often called **alignment**
    - We will get back to this very soon

Working with "Large" Language Models

- A simple way to turn LLMs into task-specific models is through fine-tuning

  - Identical to what we saw with BERT and others: fine-tune with annotated data

  - You benefit from the rich representations of the LM

- Larger models offer a completely new mode of operation that does not require any change to their parameters: **prompting**

  - With or without annotated examples: *zero-shot* or *in-context learning* (few-shot)

  - With or without intermediate reasoning steps: *chain-of-thought* prompting

Zero-shot Prompting

- Input: single unlabeled example x

- Output: the label y

- The task (and output) can be any text-to-text task: classification, summarization, translation

- Pre-processing: wrap x with a template using a **verbalizer** v(x)

- The template controls the output

x: the movie's acting could've been better, but the visuals and directing were top-notch.

↓

v(x): Review: the movie's acting could've been better, but the visuals and directing were top-notch.
Out of positive, negative, or neutral this review is

↓

LLM

↓

neutral $\bar{y}$

Zero-shot Prompting

Constrained Output

- We generate from the model to get the output

  - What if the model output does not fit the intended format, even if it is semantically correct?

    - "... how many stars on a scale of four? 4" vs.   "... how many stars on a scale of four? four stars"

  - Or maybe not even semantically correct, but just irrelevant?

Zero-shot Prompting

Constrained Output

- We generate from the model to get the output

  - What if the model output does not fit the intended format, even if it is semantically correct?

    - "… how many stars on a scale of four? 4" vs. "… how many stars on a scale of four? four stars"

- Generate with constraints:

  - Compare the probabilities of all possible outputs according to your format

$$\arg \max_{\bar{y} \in \{1,2,3,4\}} p(\bar{y} \mid v(\bar{x}))$$

Zero-shot Prompting

Constrained Output

- Generate with constraints:

  - Compare the probabilities of all possible outputs according to your format

  - If the label is a single token, just compare next token probabilities over labels

  - Otherwise?

Zero-shot Prompting

Sensitivity and Variability

- Prompting simplifies some aspects of adapting LLMs for tasks

  - No need to do expensive parameter estimate

  - You need much less data: no training data with zero-shot prompting

- However: many sources of unexpected variability

  - There are many way to write a prompt for the same task

  - Can we expect all of them to simply function the same?

Zero-shot Prompting

Sensitivity and Variability

- Minor changes that should have no impact, can have dramatic effect

- For example: asking for answer in quotations
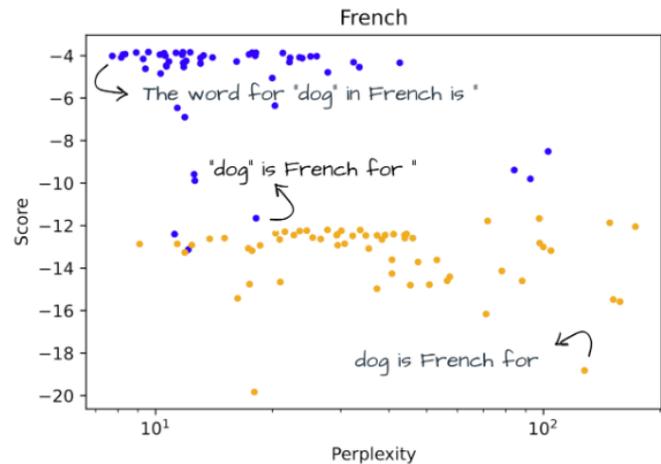
Figure 2: Score of correct label vs. perplexity for the word-level translation task in French with OPT 175B. The x axis is in log scale. The blue points stand for prompts with quotation marks for the words, while the yellow points are of prompts without quotation marks.

[Gonen et al. 2022]

Zero-shot Prompting

Sensitivity and Variability

- Prompts create a natural language input

- So the model ability to reason about that language influences task performance

  - How "natural" it is?

  - How does it "align" with the training data?

News Classification
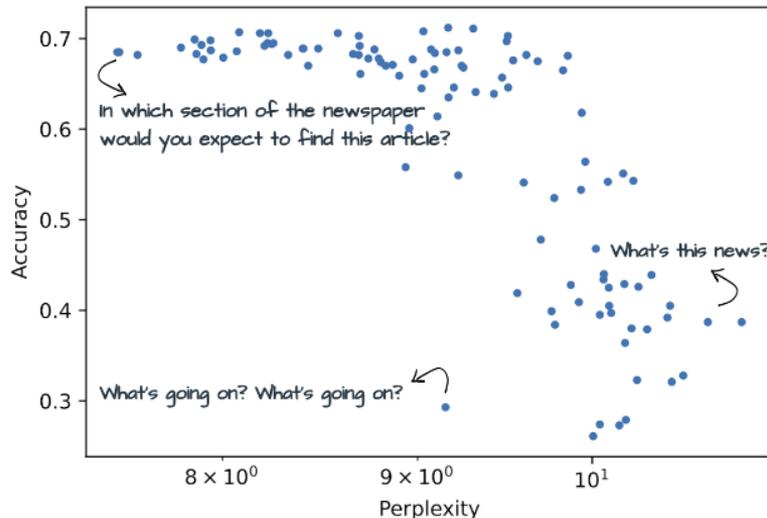
Figure 1: Accuracy vs. perplexity for the AG News dataset with OPT 175B. The x axis is in log scale. Each point stands for a different prompt.

[Gonen et al. 2022]

Zero-shot Prompting

Sensitivity and Variability

- Prompts can even be sensitive to minor cosmetic changes

- Can influence performance in unexpected ways

- Can think of them as (very complex) hyper-parameters
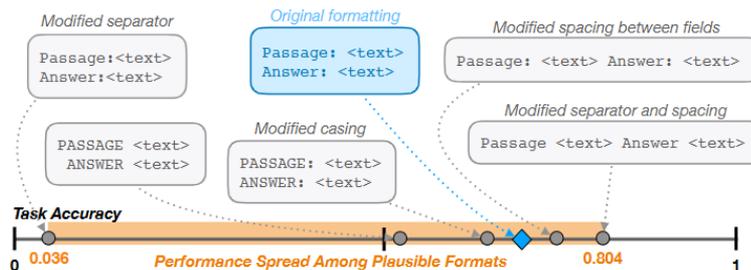
Figure 1: Slight modifications in prompt format templating may lead to significantly different model performance for a given task. Each `<text>` represents a different variable-length placeholder to be replaced with actual data samples. Example shown corresponds to 1-shot LLaMA-2-7B performances for task280 from SuperNaturalInstructions (Wang et al., 2022). This StereoSet-inspired task (Nadeem et al., 2021) requires the model to, given a short passage, classify it into one of four types of stereotype or anti-stereotype (gender, profession, race, and religion).

https://arxiv.org/abs/2310.11324

14

[Sc

Zero-shot Prompting

Prompt Optimization

- Just like hyper-parameters, can think of optimizing prompts

- There are methods for searching over prompts (either using gradients or black-box optimization)

- Most do not lead to dramatically better results compared to manual engineering/hill-climbing (and are computationally intensive)

- Most important: the choice of prompt is very important for zero-shot settings

In-context Learning (ICL)

- LLMs have the ability to "learn" to complete tasks through training in the prompt

- The recipe is simple:

  - Take a small number of annotated training example

  - Convert them using verbalizer templates

  - Concatenate them and follow with the target input

  - The completion will be the label of the input

the movie's acting could've been better, but the visuals and directing were top-notch.

⬇

Review: The cinematography was stellar; great movie! Sentiment (positive or negative): positive
Review: The plot was boring and the visuals were subpar.
Sentiment (positive or negative): negative
Review: The movie's acting could've been better, but the visuals and directing were top-notch.
Sentiment (positive or negative):

⬇

LLM

⬇

positive $\bar{y}$

In-context Learning (ICL)

Performance

- Providing ICL examples almost always leads to significant improvements

Aggregate Performance Across Benchmarks
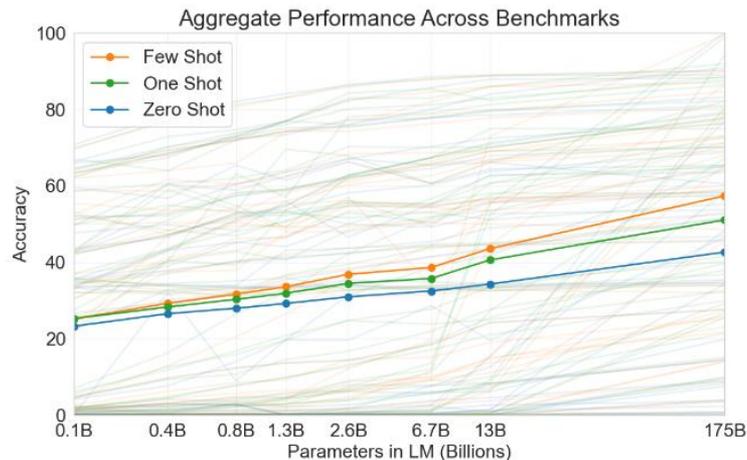
Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

[Brown et al. 2020]

In-context Learning (ICL)

Performance

- Providing ICL examples almost always leads to significant improvements

- Benefits tend to diminish with more examples
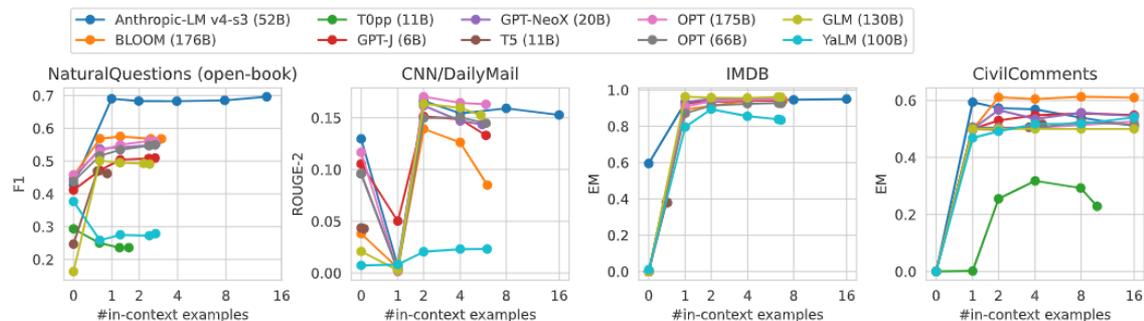
Figure 32: **Number of in-context examples.** For each model, we set the maximum number of in-context examples to [0, 1, 2, 4, 8, 16] and fit as many in-context examples as possible within the context window. We plot performance as a function of the average number of in-context examples actually used.

[Liang et al. 2022]

In-context Learning

Performance

- Model scale is important

- More examples have diminishing return

- What is the cost of more examples?
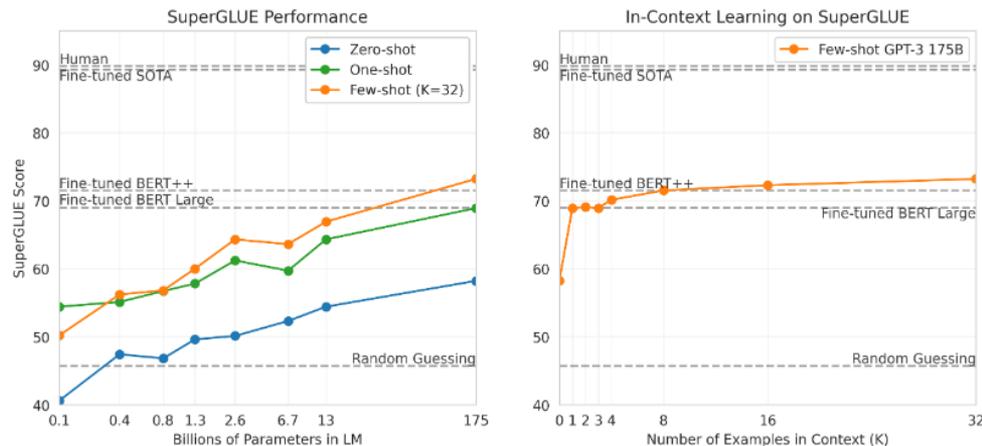
Figure 3.8: Performance on SuperGLUE increases with model size and number of examples in context. A value of $K = 32$ means that our model was shown 32 examples per task, for 256 examples total divided across the 8 tasks in SuperGLUE. We report GPT-3 values on the dev set, so our numbers are not directly comparable to the dotted reference lines (our test set results are in Table 3.8). The BERT-Large reference model was fine-tuned on the SuperGLUE training set (125K examples), whereas BERT++ was first fine-tuned on MultiNLI (392K examples) and SWAG (113K examples) before further fine-tuning on the SuperGLUE training set (for a total of 630K fine-tuning examples). We find the difference in performance between the BERT-Large and BERT++ to be roughly equivalent to the difference between GPT-3 with one example per context versus eight examples per context.

[Brown et al. 2020]

In-context Learning (ICL)

Sensitivity

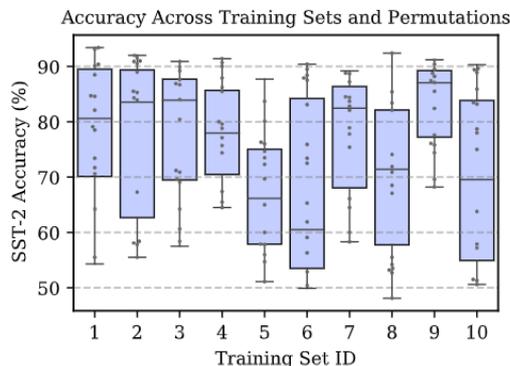- ICL can be highly sensitive to the choice of examples, their ordering, and the format of the prompt

Figure 2. There is high variance in GPT-3's accuracy as we change the prompt's **training examples**, as well as the **permutation** of the examples. Here, we select ten different sets of four SST-2 training examples. For each set of examples, we vary their permutation and plot GPT-3 2.7B's accuracy for each permutation (and its quartiles).
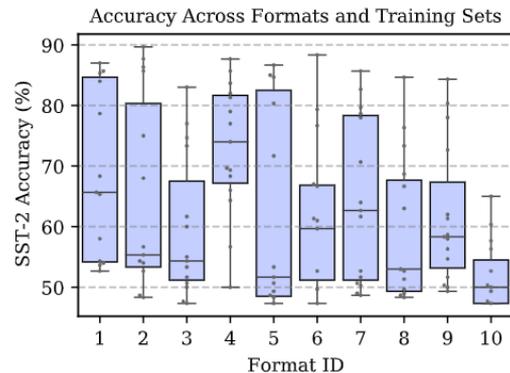
Figure 3. There is high variance in GPT-3's accuracy as we change the **prompt format**. In this figure, we use ten different prompt formats for SST-2. For each format, we plot GPT-3 2.7B's accuracy for different sets of four training examples, along with the quartiles.

In-context Learning (ICL)

Sensitivity

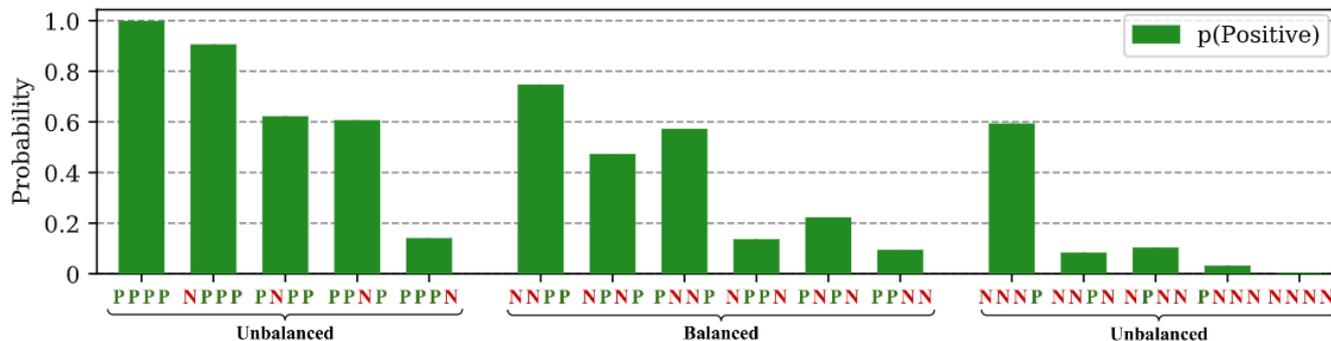- Ordering and choice of examples can lead to strong label bias

Figure 4. **Majority label and recency biases** cause GPT-3 to become biased towards certain answers and help to explain the high variance across different examples and orderings. Above, we use 4-shot SST-2 with prompts that have different class balances and permutations, e.g., [P P N N] indicates two positive training examples and then two negative. We plot how often GPT-3 2.7B predicts Positive on the balanced validation set. When the prompt is unbalanced, the predictions are unbalanced (*majority label bias*). In addition, balanced prompts that have one class repeated near the end, e.g., end with two Negative examples, will have a bias towards that class (*recency bias*).

[Zhao et al. 2021]

In-context Learning (ICL)

Sensitivity

- Particularly sensitive with fewer examples

  - Why using few examples is critical?

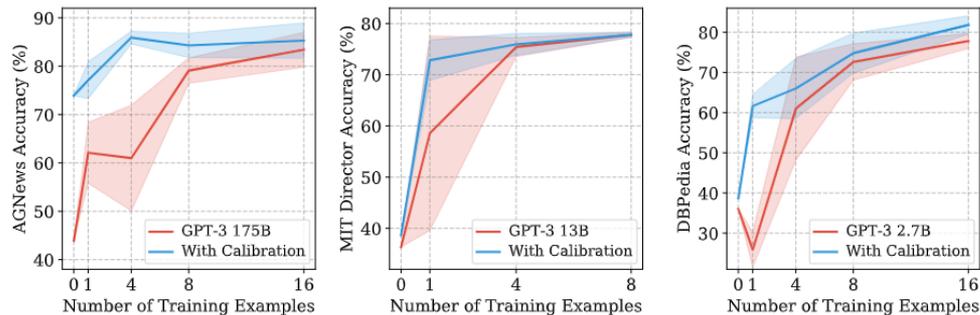- There are methods that help, for examples see [this tutorial](#)

Figure 1. Few-shot learning can be highly unstable across different choices of the prompt. Above, we plot the mean accuracy (± one standard deviation) across different choices of the training examples for three different datasets and model sizes. We show that our method, *contextual calibration*, improves accuracy, reduces variance, and overall makes tools like GPT-3 more effective for end users.

[Zhao et al. 2021]

In-context Learning (ICL)

Analysis

- In some cases, the label correctness actually matters little

- But demonstrations still important

- What's happening? Demonstration are much about domain and form
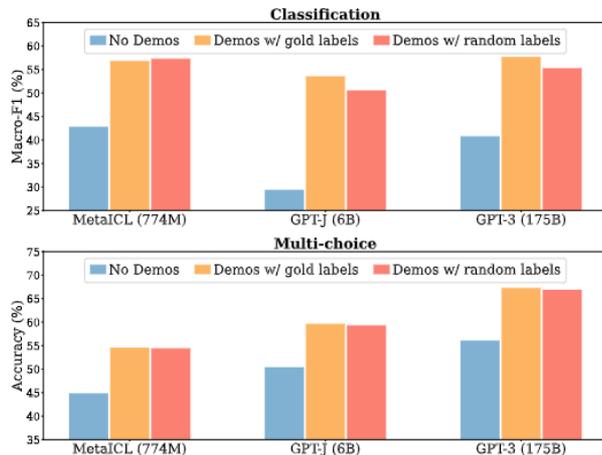
Classification

Multi-choice

Figure 1: Results in classification (top) and multi-choice tasks (bottom), using three LMs with varying size. Reported on six datasets on which GPT-3 is evaluated; the channel method is used. See Section 4 for the full results. In-context learning performance drops only marginally when labels in the demonstrations are replaced by random labels.

[Min et al. 2022]

Chain-of-thought (COT) Prompting

- Some tasks require multiple reasoning steps

- Directly generating the answer requires the model internally do the reasoning steps (or shortcut somehow)

- It is empirically useful to:

  - Show the model examples of the reasoning steps through ICL

  - And then have it explicitly generate the reasoning steps

[Wei et al. 2022]

Chain-of-thought (COT) Prompting

[Wei et al. 2022]

Chain-of-thought (COT) Prompting

Step-by-step

- COT requires ICL examples explicitly enumerating the reasoning steps

- Turn out reasoning steps can often be elicited without ICL examples

- Main idea: just "tell" the model to reason in steps

JK

Chain-of-thought (COT) Prompting

Step-by-step

(a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) The answer is 8. ✗

(b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) *The juggler can juggle 16 balls. Half of the balls are golf balls. So there are 16 / 2 = 8 golf balls. Half of the golf balls are blue. So there are 8 / 2 = 4 blue golf balls.* The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: The answer (arabic numerals) is

(Output) 8 ✗

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: **Let's think step by step.**

(Output) *There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls.* ✓

JK

Chain-of-thought (COT) Prompting

Step-by-step

- COT requires ICL examples explicitly enumerating the reasoning steps

- Turn out reasoning steps can often be elicited without ICL examples

- Main idea: just "tell" the model to reason in steps

- Challenge: the answer is often entangled in the reasoning text — how to extract it?

> Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
> A: **Let's think step by step.**
>
> (Output) *There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls.* ✓
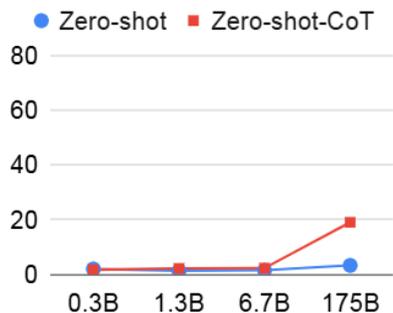
Chain-of-thought (COT) Prompting

Step-by-step

- Main idea: just "tell" the model to reason in steps

- Challenge: the answer is often entangled in the reasoning text — how to extract it? → just use an LLM 😃
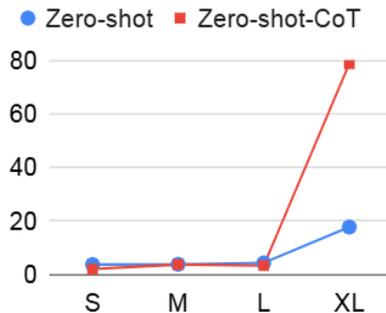
Chain-of-thought (COT) Prompting
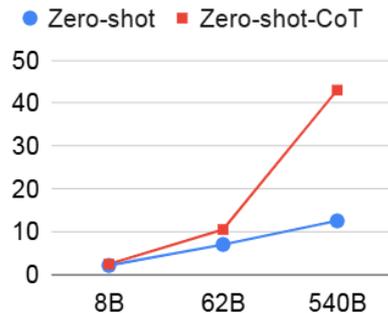
Step-by-step

- Main idea: just "tell" the model to reason in steps

- Can significantly outperform zero-shot prompting with very large models

- But requires no ICL examples

(a) MultiArith on Original GPT-3

(b) MultiArith on Instruct GPT-3

(c) GMS8K on PaLM

[Kojima et al. 2022]

Chain-of-thought (COT) Prompting

Step-by-step

- There is no one magical prompt

- Empirically, the is a set of instructive prompts that are roughly equivalent

No.	Category	Template	Accuracy
1	instructive	Let's think step by step.	**78.7**
2		First, (*1)	77.3
3		Let's think about this logically.	74.5
4		Let's solve this problem by splitting it into steps. (*2)	72.2
5		Let's be realistic and think step by step.	70.8
6		Let's think like a detective step by step.	70.3
7		Let's think	57.5
8		Before we dive into the answer,	55.7
9		The answer is after the proof.	45.7
10	misleading	Don't think. Just feel.	18.8
11		Let's think step by step but reach an incorrect answer.	18.7
12		Let's count the number of "a" in the question.	16.7
13		By using the fact that the earth is round,	9.3
14	irrelevant	By the way, I found a good restaurant nearby.	17.5
15		Abrakadabra!	15.5
16		It's a beautiful day.	13.1
-		(Zero-shot)	17.7

Table 5: Robustness study of Few-shot-CoT against examples. When the examples are from entirely different tasks, the performance generally becomes worse, but when the answer formats are matched (i.e. CommonsenseQA to AQUA-RAT, multiple-choice), the performance loss is less severe.
[†]CommonsenseQA samples are used in this variation