Best of LessWrong 2022

chinchilla's wild implications

by **nostalgebraist** 31st Jul 2022 AI Alignment Forum

(Colab notebook here.)

This post is about language model scaling laws, specifically the laws derived in the DeepMind paper that introduced Chinchilla.[1]

The paper came out a few months ago, and has been discussed a lot, but some of its implications deserve more explicit notice in my opinion. In particular:

- Data, not size, is the currently active constraint on language modeling performance. Current returns to additional data are immense, and current returns to additional model size are miniscule; indeed, most recent landmark models are wastefully big.
  - If we can leverage enough data, there is no reason to train ~500B param models, much less 1T or larger models.
  - If we *have to* train models at these large sizes, it will mean we have encountered a barrier to exploitation of data scaling, which would be a great loss relative to what would otherwise be possible.
- The literature is extremely unclear on how much text data is actually available for training. We may be "running out" of general-domain data, but the literature is too vague to know one way or the other.
- The *entire* available quantity of data in highly specialized domains like code is woefully tiny, compared to the gains that would be possible if much more such data were available.

Some things to note at the outset:

- This post assumes you have some familiarity with LM scaling laws.

- As in the paper[2], I'll assume here that models never see repeated data in training.
  - This simplifies things: we don't need to draw a distinction between data size and step count, or between train loss and test loss.
- I focus on the parametric scaling law from the paper's "Approach 3," because it's provides useful intuition.
  - Keep in mind, though, that Approach 3 yielded somewhat different results from Approaches 1 and 2 (which agreed with one another, and were used to determine Chinchilla's model and data size).
  - So you should take the exact numbers below with a grain of salt. They may be off by a few orders of magnitude (but not *many* orders of magnitude).

1. the scaling law

The paper fits a scaling law for LM loss L, as a function of model size N and data size D.

Its functional form is very simple, and easier to reason about than the $L(N, D)$ law from the earlier Kaplan et al papers. It is a sum of three terms:

$$L(N, D) = \frac{A}{N^\alpha} + \frac{B}{D^\beta} + E$$

The first term only depends on the model size. The second term only depends on the data size. And the third term is a constant.

You can think about this as follows.

An "infinitely big" model, trained on "infinite data," would achieve loss E. To get the loss for a real model, you add on two "corrections":

1. one for the fact that the model's only has N parameters, not infinitely many
2. one for the fact that the model only sees D training examples, not infinitely many

$$L(N, D) = \underbrace{\frac{A}{N^\alpha}}_{\text{finite model}} + \underbrace{\frac{B}{D^\beta}}_{\text{finite data}} + \underbrace{E}_{\text{irreducible}}$$

Here's the same thing, with the constants fitted to DeepMind's experiments on the MassiveText dataset[3].

$$L(N, D) = \underbrace{\frac{406.4}{N^{0.34}}}_{\text{finite model}} + \underbrace{\frac{410.7}{D^{0.28}}}_{\text{finite data}} + \underbrace{1.69}_{\text{irreducible}}$$

plugging in real models

Gopher is a model with 280B parameters, trained on 300B tokens of data. What happens if we plug in those numbers?

$$L(280 \cdot 10^9,\ 300 \cdot 10^9) = \underbrace{0.052}_{\text{finite model}} + \underbrace{0.251}_{\text{finite data}} + \underbrace{1.69}_{\text{irreducible}} = 1.993$$

What jumps out here is that the "finite model" term is *tiny*.

In terms of the impact on LM loss, Gopher's parameter count might as well be infinity. There's a *little* more to gain on that front, but not much.

Scale the model up to 500B params, or 1T params, or 100T params, or $3 \uparrow\uparrow\uparrow 3$ params . . . and the most this can *ever* do for you is an 0.052 reduction in loss[4].

Meanwhile, the "finite data" term is *not* tiny. Gopher's training data size is very much *not* infinity, and we can go a long way by making it bigger.

• • •

Chinchilla is a model with the same training compute cost as Gopher, allocated more evenly between the two terms in the equation.

It's 70B params, trained on 1.4T tokens of data. Let's plug that in:

$$L(70 \cdot 10^9,\ 1400 \cdot 10^9) = \underbrace{0.083}_{\text{finite model}} + \underbrace{0.163}_{\text{finite data}} + \underbrace{1.69}_{\text{irreducible}} = 1.936$$

Much better![5]

Without using any more compute, we've improved the loss by 0.057. That's bigger than Gopher's entire "finite model" term!

The paper demonstrates that Chinchilla roundly defeats Gopher on downstream tasks, as we'd expect.

Even that understates the accomplishment, though. At least in terms of loss, Chinchilla doesn't just beat Gopher. It beats *any model trained on Gopher's data, no matter how big.*

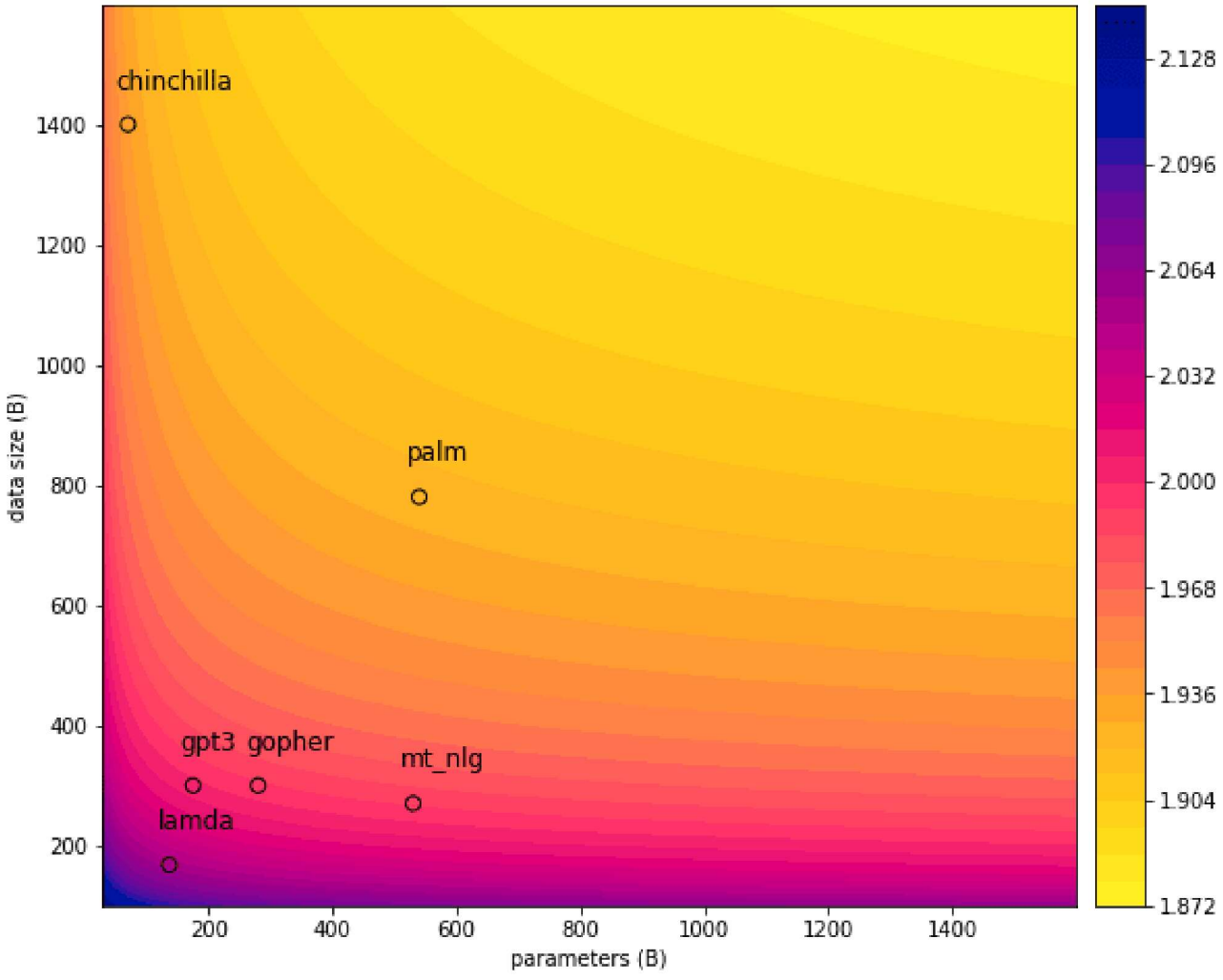
To put this in context: until this paper, it was conventional to train all large LMs on roughly 300B tokens of data. (GPT-3 did it, and everyone else followed.)

Insofar as we trust our equation, this *entire* line of research -- which includes GPT-3, LaMDA, Gopher, Jurassic, and MT-NLG -- could *never* have beaten Chinchilla, no matter how big the models got[6].

People put immense effort into training models that big, and were working on even bigger ones, and yet none of this, in principle, could ever get as far Chinchilla did.

• • •

Here's where the various models lie on a contour plot of LM loss (per the equation), with N on the x-axis and D on the y-axis.



Only PaLM is remotely close to Chinchilla here. (Indeed, PaLM does slightly better.)

PaLM is a huge model. It's the largest one considered here, though MT-NLG is a close second. Everyone writing about PaLM mentions that it has 540B parameters, and the PaLM paper does a lot of experiments on the differences between the 540B PaLM and smaller variants of it.

According to this scaling law, though, PaLM's *parameter count* is a mere footnote relative to PaLM's *training data size*.

PaLM isn't competitive with Chinchilla because it's big. MT-NLG is almost the same size, and yet it's trapped in the pinkish-purple zone on the bottom-left, with Gopher and the rest.

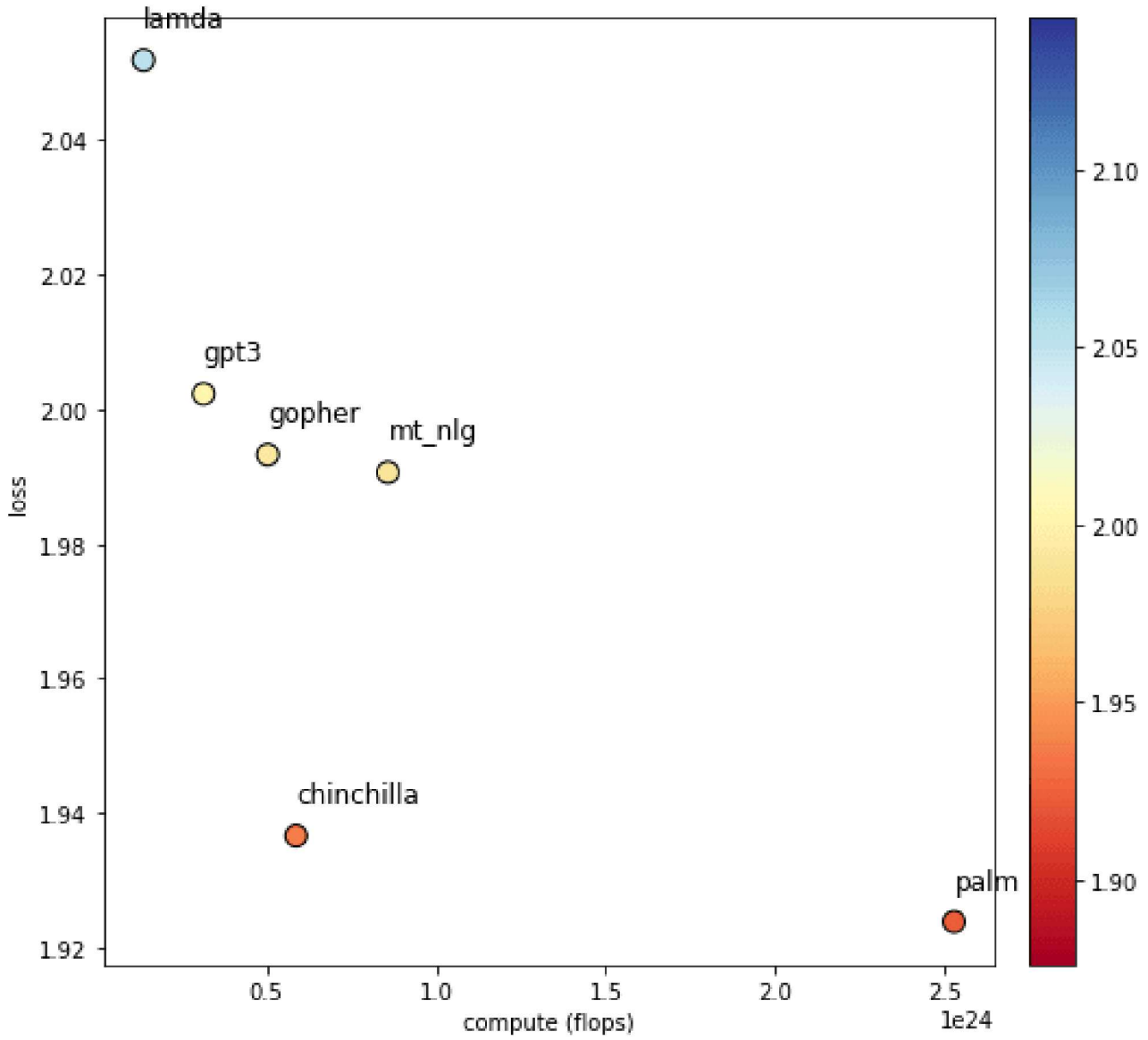
No, PaLM is competitive with Chinchilla only because it was trained on more tokens (780B) than the other non-Chinchilla models. For example, this change in data size constitutes 85% of the loss improvement from Gopher to PaLM.

Here's the precise breakdown for PaLM:

$$L(540 \cdot 10^9,\ 780 \cdot 10^9) = \underbrace{0.042}_{\text{finite model}} + \underbrace{0.192}_{\text{finite data}} + \underbrace{1.69}_{\text{irreducible}} = 1.924$$

PaLM's gains came with a great cost, though. It used way more training compute than any previous model, and its size means it also takes a lot of inference compute to run.

Here's a visualization of loss vs. training compute (loss on the y-axis and in color as well):

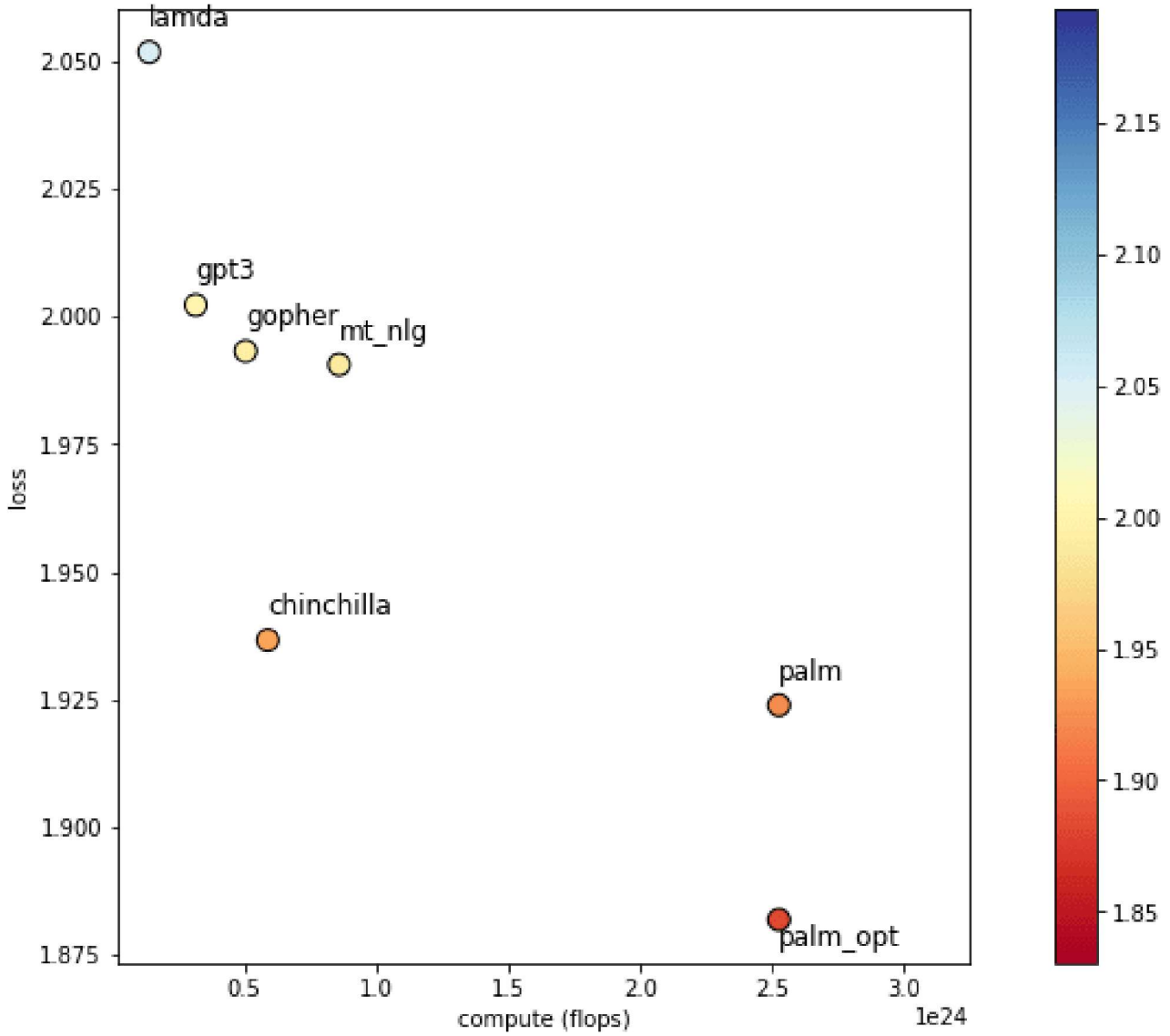


Man, we spent all that compute on PaLM, and all we got was the slightest edge over Chinchilla!

Could we have done better? In the equation just above, PaLM's terms look pretty unbalanced. Given that compute, we probably should have used more data and trained a smaller model.

The paper tells us how to pick *optimal* values for params and data, given a compute budget. Indeed, that's its main focus.

If we use its recommendations for PaLM's compute, we get the point "palm_opt" on this plot:

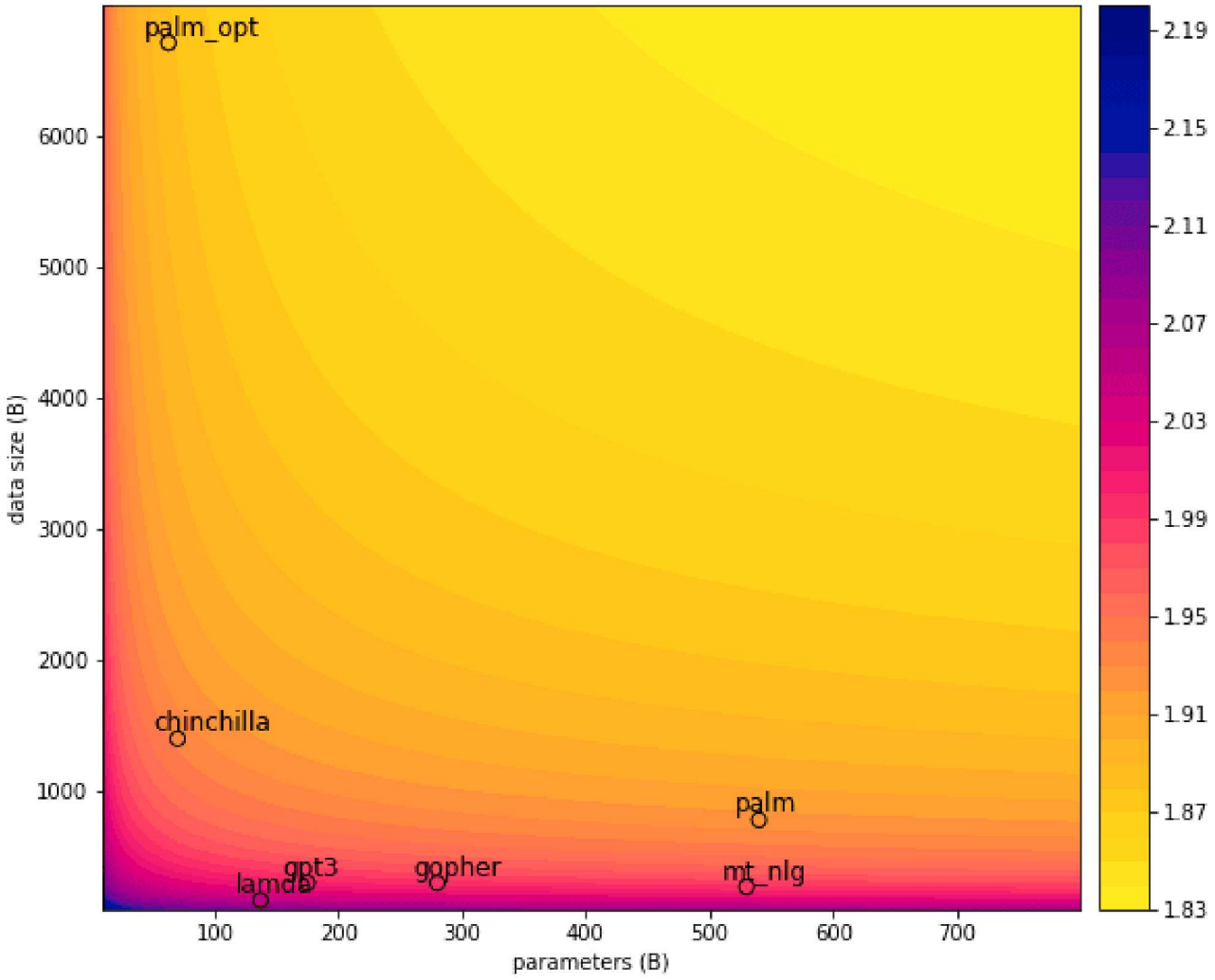


Ah, now we're talking!

• • •

"palm_opt" sure looks good. But how would we train it, concretely?

Let's go back to the *N*-vs.-*D* contour plot world.



I've changed the axis limits here, to accommodate the **massive** data set you'd need to spent PaLM's compute optimally.

How much data would that require? Around 6.7T tokens, or ~4.8 times as much as Chinchilla used.

Meanwhile, the resulting model would not be nearly as big as PaLM. The optimal compute law actually puts it at 63B params[7].

Okay, so we just need to get 6.7T tokens and . . . wait, how exactly *are* we going to get 6.7T tokens? How much text data *is* there, exactly?

2. are we running out of data?

It is frustratingly hard to find an answer to this question.

The main moral I want to get across in this post is that the large LM community has not taken data scaling seriously enough.

LM papers are meticulous about N -- doing all kinds of scaling analyses on models of various sizes, etc. There has been tons of smart discussion about the hardware and software demands of training high-N models. The question "what would it take to get to 1T params? (or 10T?)" is on everyone's radar.

Yet, meanwhile:

- Everyone trained their big models on 300B tokens, for no particular reason, until this paper showed how hilariously wasteful this is
- Papers rarely do scaling analyses that vary data size -- as if the concepts of "LM scaling" and "adding more parameters" have effectively merged in people's minds
- Papers basically never talk about what it would take to scale their *datasets* up by 10x or 50x
- The data collection sections of LM papers tend to be vague and slapdash, often failing to answer basic questions like "where did you scrape these webpages from?" or "how many more could you scrape, if you wanted to?"

As a particularly egregious example, here is what the LaMDA paper says about the composition of their training data:

> The pre-training data, called Infiniset, is a combination of dialog data from public dialog data and other public web documents. It consists of 2.97B documents and 1.12B dialogs with 13.39B utterances. The composition of the data is as follows: 50% dialogs data from public forums; 12.5% C4 data [11]; 12.5% code documents from sites related to programming like Q&A sites, tutorials, etc; 12.5% Wikipedia (English); 6.25% English web documents; and 6.25% Non-English web documents. The total number of words in the dataset is 1.56T.

"Dialogs data from public forums"? Which forums? Did you use all the forum data you could find, or only 0.01% of it, or something in between? And why measure *words* instead of tokens -- unless they *meant* tokens?

If people were as casual about scaling N as this quotation is about scaling D, the methods sections of large LM papers would all be a few sentences long. Instead, they tend to look like this (excerpted from ~3 pages of similar material):

We scale training beyond a single TPU v4 Pod using the Pathways system (Barham et al., 2022). PaLM 540B utilizes the client-server architecture of Pathways to achieve two-way data parallelism at the pod level. Here a single Python client dispatches half of the training batch to each pod, each pod executes the forward and backward computation to compute gradients in parallel using standard within-pod data and model parallelism. The pods then transfer the gradients (computed on their half of the batch) with the remote pod, and finally, each pod accumulates the local and remote gradients and applies parameter updates in parallel to obtain bitwise-identical parameters for the next timestep.

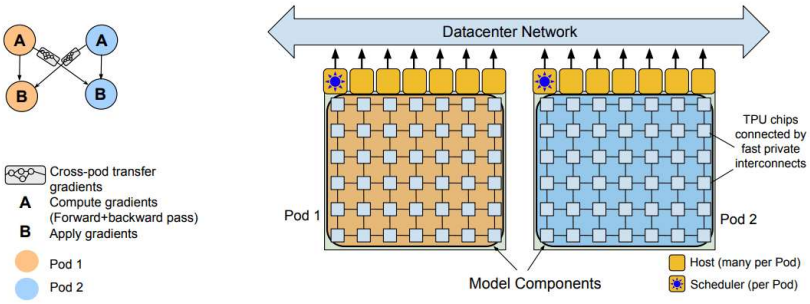


Figure 2: The Pathways system (Barham et al., 2022) scales training across two TPU v4 pods using two-way data parallelism at the pod level.

Figure 2 shows how the Pathways system executes the two-way pod-level data parallelism. A single Python client constructs a sharded dataflow program (shown on the left in Figure 2) that launches JAX/XLA (XLA, 2019) work on remote servers that each comprise a TPU pod. The program contains a component A for within-pod forward+backward computation (including within-pod gradient reduction), transfer subgraph for cross-pod gradient transfer, and a component B for optimizer update (including summation of local and remote gradients). The Pathways program executes component A on each pod, then transfers the output

From the PaLM paper

• • •

...anyway. How much more data could we get?

This question is complicated by the fact that not all data is equally good.

(This messy Google sheet contains the calculations behind some of what I say below.)

web scrapes

If you just want *a lot of text*, the easiest way to get it is from web scrapes like Common Crawl.

But these are infamously full of garbage, and if you want to train a good LM, you probably want to aggressively filter them for quality. And the papers don't tell us how much *total* web data they have, only how much *filtered* data.

MassiveWeb

The training dataset used for Gopher and Chinchilla is called MassiveText, and the web scrape portion of it is called MassiveWeb. This data originates in a mysterious, unspecified web scrape[8], which is funneled through a series of filters, including quality heuristics and an attempt to only keep English text.

MassiveWeb is 506B. Could it be made bigger, by scaling up the original web scrape? That depends on how complete the original web scrape was -- but we know nothing about it.

The GLaM/PaLM web corpus

PaLM used a different web scrape corpus. It was first used in this paper about "GLaM," which again did not say anything about the original scraping process, only describing the quality filtering they did (and not in much detail).

The GLaM paper says its filtered web corpus is 143B tokens. That's a lot smaller than MassiveWeb. Is that because of the filtering? Because the original scrape was smaller? Dunno.

To further complicate matters, the PaLM authors used a *variant* of the GLaM dataset which made multilingual versions of (some of?) the English-only components.

How many tokens did this add? They don't say[9].

We *are* told that 27% (211B) of PaLM's training tokens came from this web corpus, and we are separately told that they tried to avoid repeating data. So the PaLM version of the GLaM web corpus is probably at least 211B, versus the original 143B. (Though I am not very confident of that.)

Still, that's much smaller than MassiveWeb. Is this because they had a higher quality bar (which would be bad news for further data scaling)? They do attribute some of PaLM's success to quality filtering, citing the ablation on this in the GLaM paper[10].

It's hard to tell, but there is this ominous comment, in the section where they talk about PaLM vs. Chinchilla:

> Although there is a large amount of very high-quality textual data available on the web, there is not an infinite amount. For the corpus mixing proportions chosen for PaLM, data begins to repeat in some of our subcorpora after 780B tokens, which is why we chose that as the endpoint of training. It is unclear how the "value" of repeated data compares to unseen data for large-scale language model training[11].

The subcorpora that start to repeat are probably the web and dialogue ones.

Read literally, this passage seems to suggest that even the vast web data resources available to *Google Research* (!) are starting to strain against the data demands of large

LMs. Is that plausible? I don't know.

domain-specific corpora

We can speak with more confidence about text in specialized domains that's less common on the open web, since there's less of it out there, and people are more explicit about where they're getting it.

Code

If you want code, it's on Github. There's some in other places too, but if you've exhausted Github, you probably aren't going to find orders of magnitude of additional code data. (I think?)

We've more-or-less exhausted Github. It's been scraped a few times with different kinds of filtering, which yielded broadly similar data sizes:

- The Pile's scrape had 631GB[12] of text, and ~299B tokens
- The MassiveText scrape had 3.1TB of text, and 506B tokens
- The PaLM scrape had only 196GB of text (we aren't told how many tokens)
- The Codex paper's scrape was python-only and had 159GB of text

(The text to token ratios vary due to differences in how whitespace was tokenized.)

All of these scrapes contained a large fraction of the total code available on Github (in the Codex paper's case, just the python code).

Generously, there might be **~1T tokens** of code out there, but not vastly more than that.

Arxiv

If you want to train a model on advanced academic research in physics or mathematics, you go to Arxiv.

For example, Arxiv was about half the training data for the math-problem-solving LM Minerva.

We've exhausted Arxiv. Both the Minerva paper and the Pile use basically all of Arxiv, and it amounts to a measly **21B tokens**.

Books

Books? What exactly are "books"?

In the Pile, "books" means the Books3 corpus, which means "all of Bibliotik." It contains 196,640 full-text books, amounting to only **27B tokens.**

In MassiveText, a mysterious subset called "books" has **560B tokens**. That's a lot more than the Pile has! Are these all the books? In . . . the world? In . . . Google books? Who even knows?

In the GLaM/PaLM dataset, an equally mysterious subset called "books" has **390B tokens.**

Why is the GLaM/PaLM number so much smaller than the MassiveText number? Is it a tokenization thing? Both of these datasets were made by Google, so it's not like the Gopher authors have special access to some secret trove of forbidden books (I assume??).

If we want LMs to learn the kind of stuff you learn from books, and not just from the internet, this is what we have.

As with the web, it's hard to know what to make of it, because we don't know whether this is "basically all the books in the world" or just some subset that an engineer pulled at one point in time[13].

"all the data we have"

In my spreadsheet, I tried to make a rough, erring-on-generous estimate of what you'd get if you pooled together all the sub-corpora mentioned in the papers I've discussed here.

I tried to make it an overestimate, and did some extreme things like adding up both MassiveWeb *and* the GLaM/PaLM web corpus as though they were disjoint.

The result was **~3.2T** tokens, or

- about 1.6x the size of MassiveText
- about 35% of the data we would need to train palm_opt

Recall that this already contains "basically all" of the open-source code in the world, and "basically all" of the theoretical physics papers written in the internet era -- within an order of magnitude, anyway. In these domains, the "low-hanging fruit" of data scaling are not low-hanging at all.

what is compute? (on a further barrier to data scaling)

Here's another important comment from the PaLM paper's Chinchilla discussion. This is about barriers to doing a head-to-head comparison experiment:

> If the smaller model were trained using fewer TPU chips than the larger model, this would proportionally increase the wall-clock time of training, since the total training FLOP count is the same. If it were trained using the same number of TPU chips, it would be very difficult to maintain TPU compute efficiency without a drastic increase in batch size. The batch size of PaLM 540B is already 4M tokens, and it is unclear if even larger batch sizes would maintain sample efficiency.

In LM scaling research, all "compute" is treated as fungible. There's one resource, and you spend it on params and steps, where compute = params * steps.

But params can be *parallelized*, while steps cannot.

You can take a big model and spread it (and its activations, gradients, Adam buffers, etc.) across a cluster of machines in various ways. This is how people scale up N in practice.

But to scale up D, you have to either:

- take more optimization steps -- an inherently serial process, which takes linearly more time as you add data, no matter how fancy your computers are
- increase the batch size -- which tends to degrade model quality beyond a certain critical size, and current high-N models are already pushing against that limit

Thus, it is unclear whether the "compute" you spend in high-D models is as readily available (and as bound to grow over time) as we typically imagine "compute" to be.

If LM researchers start getting serious about scaling up data, no doubt people will think hard about this question, but that work has not yet been done.

appendix: to infinity

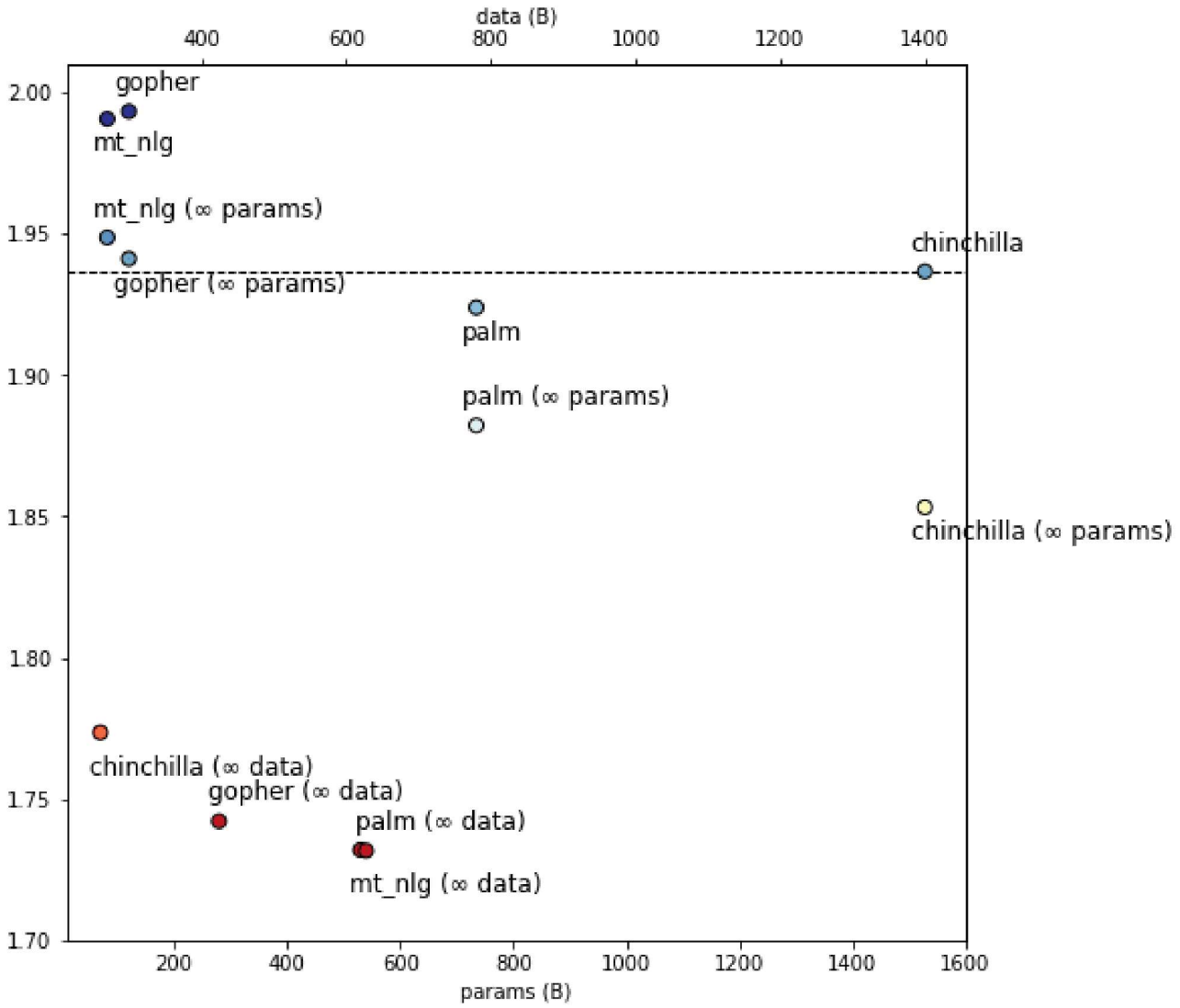
Earlier, I observation that Chinchilla beats any Gopher of arbitrary size.

The graph below expands on that observation, by including two variants of each model:

- one with the finite-model term set to zero, i.e. the infinite-parameter limit
- one with the finite-data term set to zero, i.e. the infinite-data limit

(There are two x-axes, one for data and one for params. I included the latter so I have a place to put the infinite-data models without making an infinitely big plot.

The dotted line is Chinchilla, to emphasize that it beats infinite-params Gopher.)



The main takeaway IMO is the size of the gap between ∞ data models and all the others. Just another way of emphasizing how skewed these models are toward N, and away from D.

---

1. ^ Training Compute-Optimal Large Language Models
2. ^ See their footnote 2

3. ^ See their equation (10)

4. ^ Is 0.052 a "small" amount in some absolute sense? Not exactly, but (A) it's small compared to the loss improvements we're used to seeing from new models, and (B) small compared to the improvements possible by scaling data.

   In other words, (A) we have spent a few years plucking low-hanging fruit much bigger than this, and (B) there are more such fruit available.

5. ^ The two terms are still a bit imbalanced, but that's largely due to the "Approach 3 vs 1/2" nuances mentioned above.

6. ^ Caveat: Gopher and Chinchilla were trained on the same data distribution, but these other models were not. Plugging them into the equation won't give us accurate loss values for the datasets they used. Still, the datasets are close enough that the broad trend ought to be accurate.

7. ^ Wait, isn't that *smaller* than Chinchilla?

   This is another Approach 3 vs. 1/2 difference.

   Chinchilla was designed with Approaches 1/2. Using Approach 3, like we're doing here, give you a Chinchilla of only 33B params, which *is* lower than our palm_opt's 63B.

8. ^ Seriously, I can't find *anything* about it in the Gopher paper. Except that it was "collected in November 2020."

9. ^ It is not even clear that this multilingual-ization affected the web corpus at all.

   Their datasheet says they "used multilingual versions of Wikipedia and conversations data." Read literally, this would suggest they *didn't* change the web corpus, only those other two.

   I also can't tell if the original GLaM web corpus was English-only to begin with, since that paper doesn't say.

10. ^ This ablation only compared filtered web data to *completely* unfiltered web data, which is not a very fine-grained signal. (If you're interested, EleutherAI has done more extensive experiments on the impact of filtering at smaller scales.)

11. ^ They are being a little coy here. The current received wisdom by now is that repeating data is *really bad* for LMs and you should never do it. See this paper and this one.
    EDIT 11/15/22: but see also the Galactica paper, which casts significant doubt on this claim.

12. ^ The Pile authors only included a subset of this in the Pile.

13. ^ The MassiveText datasheet says only that "the books dataset contains books from 1500 to 2008," which is not especially helpful.

Mentioned in

340 Why I think strong general AI is coming soon

146 Algorithmic Improvement Is Probably Faster Than Scaling Now
145 Yudkowsky vs Hanson on FOOM: Whose Predictions Were Better?
112 GPT-4 Predictions
80 What's the Least Impressive Thing GPT-4 Won't be Able to Do
Load More (5/26)

Rendering 127/128 comments, sorted by top scoring (show more)

Some comments are truncated due to high volume. (⌘F to expand all) Change truncation settings

[-] **Ivan Vendrov** 3y ▼ 96 ▲ ✕ 19 ✓

Thought-provoking post, thanks.

One important implication is that pure AI companies such as OpenAI, Anthropic, Conjecture, Cohere are likely to fall behind companies with access to large amounts of non-public-internet text data like Facebook, Google, Apple, perhaps Slack. Email and messaging are especially massive sources of "dark" data, provided they can be used legally and safely (e.g. without exposing private user information). Taking just email, something like 500 billion emails are sent daily, which is more text than any LLM has ever been trained on (admittedly with a ton of duplication and low quality content).

Another implication is that federated learning, data democratization efforts, and privacy regulations like GDPR are much more likely to be critical levers on the future of AI than previously thought.

[-] **Thirkle** 3y ▼ 42 ▲ ✕ 10 ✓

Another implication is that centralised governments with the ability to aggressively collect and monitor citizen's data, such as China, could be major players.

A government such as China has no need to scrape data from the Internet, while being mindful of privacy regulations and copyright. Instead they can demand 1.4 billion people's data from all of their domestic tech companies. This includes everything such as emails, texts, WeChat, anything that the government desires.

8 **Yitz** 3y I suspect that litigation over copyright concerns with LLMs could significantly slow ti...
5 **hackpert** 3y I mean Microsoft for one seems fully invested in (married to) OpenAI and will c...
9 **ChristianKl** 3y Allowing OpenAI to use Microsofts customer data to train the model essen...
5 **[anonymous]** 3y And presumably data poisoning as well? This sort of thing isn't easily influ...

[-] **Scott Alexander** 3y ▼ 63 ▲ ✕ 0 ✓

Thanks for posting this, it was really interesting. Some very dumb questions from someone who doesn't understand ML at all:

1. All of the loss numbers in this post "feel" very close together, and close to the minimum loss of 1.69. Does loss only make sense on a very small scale (like from 1.69 to 2.2), or is this telling us that language models are very close to optimal and there are only minimal remaining possible gains? What was the loss of GPT-1?

2. Humans "feel" better than even SOTA language models, but need less training data than those models, even though right now the only way to improve the models is through more training data. What am I supposed to conclude from this? Are humans running on such a different paradigm that none of this matters? Or is it just that humans are better at common-sense language tasks, but worse at token-prediction language tasks, in some way where the tails come apart once language models get good enough?

3. Does this disprove claims that "scale is all you need" for AI, since we've already maxed out scale, or are those claims talking about something different?

[-] **nostalgebraist** 3y ▼ 47 ▲ ✕ 7 ✓

(1)

Loss values are useful for comparing different models, but I don't recommend trying to interpret what they "mean" in an absolute sense. There are various reasons for this.

One is that the "conversion rate" between loss differences and ability differences (as judged by humans) changes as the model gets better and the abilities become less trivial.

Early in training, when the model's progress looks like realizing "*huh, the word 'the' is more common than some other words*", these simple insights correspond to relatively large decreases in loss. Once the model basically kinda knows English or whatever the language is, it's already made most of the loss progress it's going to make, and the further insights we really care about involve much smaller changes in loss. See here for more on this by gwern.

(2)

No one really knows, but my money is on "humans are actually better at this through some currently-unknown mechanism," as opposed to "humans are actually bad at this exact thing."

Why do I think this?

Well, the reason we're here talking about this at all is that LMs *do* write text of spookily high quality, even if they aren't as good as humans at it. That wasn't always... (read more)

[-] **Buck** 3y* ▼ 34 ▲ ✕ 3 ✓

> *That is, I suspect humans could be trained to perform very well, in the usual sense of "training" for humans where not too much data/time is necessary.*

I paid people to try to get good at this game, and also various smart people like Paul Christiano tried it for a few hours, and everyone was still notably worse than GPT-2-sm (about the size of GPT-1).

EDIT: These results are now posted here°.

[-] **paulfchristiano** 3y ▼ 34 ▲ ✕ 6 ✓

I expect I would improve significantly with additional practice (e.g. I think a 2nd hour of playing the probability-assignment game would get a much higher score than my 1st in expectation). My subjective feeling was that I could probably learn to do as well as GPT-2-small (though estimated super noisily) but there's definitely no way I was going to get close to GPT-2.

[-] **nostalgebraist** 3y ▼ 16 ▲ ✕ 3 ✓

I'm wary of the assumption that we can judge "human ability" on a novel task X by observing performance after an hour of practice.

There are *some* tasks where performance improves with practice but plateaus within one hour. I'm thinking of relatively easy video games. Or relatively easy games in general, like casual card/board/party games with simple rules and optimal policies. But most interesting things that humans "can do" take much longer to learn than this.

Here are some things that humans "can do," but require >> 1 hour of practice to "do," while still requiring far less exposure to task-specific example data than we're used to in ML:

- Superforecasting
  - Reporting calibrated numeric credences, a prerequisite for both superforecasting and the GPT game (does this take >> 1 hour? I would guess so, but I'm not sure)
- Playing video/board/card games of nontrivial difficulty or depth
- Speaking any given language, even when learned during the critical language acquisition period
- Driving motor vehicles like cars (arguably) and planes (definitely)
- Writing good prose, for any conventional sense of "good" in any genre/style
- Juggling
- Computer programming (with any prof

... (read more)

[-] **Buck** 3y ▼ 11 ▲ ✕ 1 ✓

Ok, sounds like you're using "not too much data/time" in a different sense than I was thinking of; I suspect we don't disagree. My current guess is that some humans could beat GPT-1 with ten hours of practice, but that GPT-2 or larger would be extremely difficult or and plausibly impossible with any amount of practice.

4 **jacob_cannell** 3y The human brain internally is performing very similar computations ...

4 **Owain_Evans** 3y It could be useful to look at performance of GPT-3 on foreign languages...

3 **Mateusz Bagiński** 2y I think I remember seeing somewhere that LLMs learn more slowl...

[-] **Beth Barnes** 3y ▼ 22 ▲ ✕ 0 ✓

Based on the language modeling game that Redwood made, it seems like humans are much worse than models at next word prediction (maybe around the performance of a 12-layer model)

[-] **iceman** 3y ▼ 12 ▲ ✕ 0 ✓

> What changed with the transformer? To some extent, the transformer is really a "smarter" or "better" architecture than the older RNNs. If you do a head-to-head comparison with the same training data, the RNNs do worse.

> But also, it's feasible to scale transformers much bigger than we could scale the RNNs. You don't see RNNs as big as GPT-2 or GPT-3 simply because it would take too much compute to train them.

You might be interested in looking at the progress being made on the RWKV-LM architecture, if you aren't following it. It's an attempt to train an RNN like a transformer. Initial numbers look pretty good.

3 **Owain_Evans** 3y A few points: 1. Current models do pretty well on tricky math problems (...

1 **Jose Miguel Cruz y Celis** 3y I'm curious about where you get that "models trained mostl...

[-] **wickemu** 3y ▼ 11 ▲ ✕ -3 ✓

> 2. Humans "feel" better than even SOTA language models, but need less training data than those models, even though right now the only way to improve the models is through more training data. What am I supposed to conclude from this? Are humans running on such a different paradigm that none of this matters? Or is it just that humans are better at common-sense language tasks, but worse at token-prediction language tasks, in some way where the tails come apart once language models get good enough?

Why do we say that we need less training data? Every minute instant of our existence is a multisensory point of data from before we've even exited the womb. We spend months, arguably years, hardly capable of anything at all yet still taking and retaining data. Unsupervised and mostly redundant, sure, but certainly not less than a curated collection of Internet text. By the time we're teaching a child to say "dog" for the first time they've probably experienced millions of fragments of data on creatures of various limb quantities, hair and fur types, sizes, sounds and smells, etc.; so they're already effectively pretrained on animals before we first provide a supervised connection between the sound "dog" and the sight of a four-limbed hairy creature with long ears on a leash.

I believe that Humans exceed the amount of data ML models have by multiple orders of magnitude by the time we're adults, even if it's extremely messy.

4 **Jose Miguel Cruz y Celis** 3y I did some calculations with a bunch of assumptions and simpl...

4 **Lukas Finnveden** 3y There's a billion seconds in 30 years. Chinchilla was trained on 1.4 tril...

2 **ChristianKl** 3y That depends a lot on how you count. A quick Googling suggest that the o...

2 **Lukas Finnveden** 3y (If 1 firing = 1 bit, that should be 34 megabit ~= 4 megabyte.) This ...

[-] **IL** 3y Ω 17 ▼ 63 ▲ ✕ 19 ✓

When you exhaust all the language data from text, you can start extracting language from audio and video.

As far as I know the largest public repository of audio and video is YouTube. We can do a rough back-of-the-envelope computation for how much data is in there:

- According to some 2019 article I found, in every minute 50 hours of video are uploaded to YouTube. If we assume this was the average for the last 15 years, that gets us 200 billion minutes of video.
- An average conversation has 150 words per minute, according to a Google search. That gets us 30T words, or 30T tokens if we assume 1 token per word (is this right?)

- Let's say 1% of that is actually useful, so that gets us 300B tokens, which is... a lot less than I expected.

So it seems like video doesn't save us, if we just use it for the language data. We could do self-supervised learning on the video data, but for that we need to know the scaling laws for video (has anyone done that?).

[-] **nostalgebraist** 3y Ω 5 ▼ 21 ▲ ✕ 10 ✓

Very interesting!

There are a few things in the calculation that seem wrong to me:

- If I did things right,15 years * (365 days/yr) * (24 hours/day) * (60 mins/hour) * (50 youtube!hours / min) * (60 youtube!mins / youtube!hour) = 24B youtube!minutes, not 200B.
- I'd expect much less than 100% of Youtube video time to contain speech. I don't know what a reasonable discount for this would be, though.
- In the opposite direction, 1% useful seems too low. IIRC, web scrape quality pruning discards less than 99%, and this data is less messy than a web scrape.

In any case, yeah, this does not seem like a huge amount of data. But there's enough order-of-magnitude fuzziness in the estimate that it does seem like it's worth someone's time to look into more seriously.

[-] **Sam Bowman** 3y Ω 7 ▼ 17 ▲ ✕ 16 ✓

I agree that this points in the direction of video becoming increasingly important.

But why assume only 1% is useful? And more importantly, why use only the language data? Even if we don't have the scaling laws, but it seems pretty clear that there's a ton of information in the non-language parts of videos that'd be useful to a general-purpose agent—almost certainly more than in the language parts. (Of course, it'll take more computation to extract the same amount of useful information from video than from text.)

[-] **MSRayne** 3y Ω 17 ▼ 54 ▲ ✕ 15 ✓

Does this imply that AGI is not as likely to emerge from language models as might have been thought? To me it looks like it's saying that the only way to get enough data would be to have the AI actively interacting in the world - getting data *itself*.

[-] **nostalgebraist** 3y Ω 19 ▼ 72 ▲ ✕ 14 ✓

I definitely think it makes LM --> AGI less likely, although I didn't think it was very likely to begin with °.

I'm not sure that the AI interacting with the world would help, at least with the narrow issue described here.

If we're talking about data produced by humans (perhaps solicited from them by an AI), then we're limited by the timescales of human behavior. The data sources described in this post were produced by millions of humans writing text over the course of decades (in rough order-of-magnitude terms).

All that text was already there in the world when the current era of large LMs began, so large LMs got to benefit from it immediately, "for free." But once it's exhausted, producing more is slow.

• • •

IMO, most people are currently overestimating the potential of large generative models -- including image models like DALLE2 -- because of this fact.

There was all this massive data already sitting around from human activity (the web, Github, "books," Instagram, Flickr, etc) long before ML compute/algorithms were anywhere near the point where they needed *more* data than that.

When our compute finally began to catch up with our data, we effectively spent all the "stored-up p... (read more)

7 **MSRayne** 3y It seems to me that the key to human intelligence is nothing like what LMs d...

6 **clone of saturn** 3y Language models seem to do a pretty good job at judging text "quality"...

4 **Vladimir_Nesov** 3y It might be even better to just augment the data with quality judge...

4 **Evan R. Murphy** 3y We may be running up against text data limits on the public web. But t...

[-] **MathiasKB** 3y ▼ 12 ▲ ✕ 4 ✓

I don't think the real world is good enough either.

The fact that humans feel a strong sense of the tetris effect, suggest to me that the brain is constantly generating and training on synthetic data.

5 **Yitz** 3y Aka dreams?

[-] **Roman Leventov** 3y ▼ 24 ▲ ✕ 10 ✓

My two cents contra updates towards longer or more uncertain AGI timelines given the information in this post:

- The training of language models is many orders of magnitude less efficient than the training of the human brain, which acquires comparable language comprehension and generation ability on a tiny fraction of the text corpora discussed in this post. So we can expect more innovations that improve the training efficiency. Even one such innovation, improving the training efficiency (in terms of data) by a single order of magnitude, would probably ensure that the total size of publically available text data is *not* a roadblock on the path to AGI, even if it is, currently. I think the probability that we will see at least one such innovation in the next 5 years is quite high, more than 10%.
- Perhaps DeepMind's Gato is already a response to the realisation that "there is not enough text", explained in this post. So they train Gato on game replays, themselves generated programmatically, using RL agents. They can generate practically unlimited amounts of training data in this way. Then there is probably a speculation that at some scale, Gato will generalise the knowledge acquired in games to text, or will indeed enable much more efficient training on text, (a-la few-shot learning in current LMs) if the model is pre-trained on games and other tasks.

[-] **Jay Bailey** 3y ▼ 23 ▲ ✕ 0 ✓

I am curious about this "irreducible" term in the loss. Apologies if this is covered by the familiarity with LM scaling laws mentioned as a prerequisite for this article.

When you say "irreducible", does that mean "irreducible under current techniques" or "mathematically irreducible", or something else?

Do we have any idea what a model with, say, 1.7 loss (i.e, a model almost arbitrarily big in compute and data, but with the same 1.69 irreducible) would look like?

[-] **nostalgebraist** 3y ▼ 15 ▲ ✕ 1 ✓

> When you say "irreducible", does that mean "irreducible under current techniques" or "mathematically irreducible", or something else?

Closer to the former, and even more restrictive: "irreducible with this type of model, trained in this fashion on this data distribution."

Because language is a communication channel, there is presumably *also* some nonzero lower bound on the loss that *any* language model could ever achieve. This is different from the "irreducible" term here, and presumably lower than it, although little is known about this issue.

> Do we have any idea what a model with, say, 1.7 loss (i.e, a model almost arbitrarily big in compute and data, but with the same 1.69 irreducible) would look like?

Not really, although section 5 of this post° expresses some of my own intuitions about what this limit looks like.

Keep in mind, also, that we're talking about LMs trained on a specific data distribution, and only evaluating their loss on data sampled from that same distribution.

So if an LM achieved 1.69 loss on MassiveText (or a scaled-up corpus that looked like MassiveText in all respects but size), it would do very well at mimicking all the types of text present in MassiveText, but that does not mean it could mimic every existing kind of text (much less every *conceivable* kind of text).

1 **Yitz** 3y Do we have a sense of what the level of loss is in the human brain? If I'm understa...

2 **Lone Pine** 3y Theroetically we could measure it by having humans play "the language m...

[-] **Yitz** 3y ▼ 15 ▲ ✕ 0 ✓

Such a game already exists! See https://rr-lm-game.herokuapp.com/whichonescored2 and https://rr-lm-game.herokuapp.com/. I've been told humans tend to do pretty badly at the games (I didn't do too well myself), so if you feel discouraged playing and want a similar style of game that's perhaps a bit more fun (if slightly less relevant to the question at hand), I recommend https://www.redactle.com/. Regardless, I guess I'm thinking of loss (in humans) in the more abstract sense of "what's the distance between the correct and human-given answer [to an arbitrary question about the real world]?" If there's some mathematically necessary positive amount of loss humans must have at a minimum, that would seemingly imply that there are fundamental limits to the ability of human cognition to model reality.

7 **Buck** 3y Yes, humans are way worse than even GPT-1 at next-token prediction, even a...

3 **Yitz** 3y Is there some reasonable-ish way to think about loss in the domain(s) that h...

4 **JBlack** 3y The scoring for that first game is downright bizarre. The optimal strategy for...

6 **Buck** 3y (I run the team that created that game. I made the guess-most-likely-next-t...

2 **Yitz** 3y Yeah, if anyone builds a better version of this game, please let me know!

[-] **Julian Schrittwieser** 3y ▼ 22 ▲ ✕ 6 ✓

An important distinction here is that the number of tokens a model was trained for should not be confused with the number of tokens in a dataset: if each token is seen exactly once during training then it has been trained for one "epoch".

In my experience scaling continues for quite a few epochs over the same datset, only if the model has more parameters than the datset tokens and training for >10 epochs does overfitting kick in and scaling break down.

[-] **nostalgebraist** 3y ▼ 30 ▲ ✕ 2 ✓

This distinction exists in general, but it's irrelevant when training sufficiently large LMs.

It is well-established that repeating data during large LM training is not a good practice. Depending on the model size and the amount of repeating, one finds that it is either

1. a suboptimal use of compute (relative to training a bigger model for 1 epoch), or
2. actively harmful, as measured by test loss or loss on out-of-distribution data

with (2) kicking in earlier (in terms of the amount of repeating) for larger models, as shown in this paper (Figure 4 and surrounding discussion).

For more, see

- references linked in footnote 11° of this post, on how repeating data can be harmful
- my earlier post here°, on how repeating data can be compute-inefficient even when it's not harmful
- this report on my own experience finetuning a 6.1B model, where >1 epoch was harmful

[-] **p.b.** 3y ▼ 16 ▲ ✕ 5 ✓

I think it would be a great follow-up post to explain why you think repeating data is not going to be the easy way out for the scaling enthusiasts at Deepmind and OpenAI.

I find the Figure 4 discussion at your first link quite confusing. They study repeated data i.e. disbalanced datasets to then draw conclusions about repeating data i.e. training for several epochs. The performance hit they observe seems to not be massive (when talking about scaling a couple of OOMs) and they keep the number of training tokens constant.

I really can't tell how this informs me about what would happen if somebody tried to scale compute 1000-fold and had to repeat data to do it compute-optimally, which seems to be the relevant question.

8 **ErickBall** 3y So do you think, once we get to the point where essentially all new language ...

[-] **nostalgebraist** 3y ▼ 16 ▲ ✕ 3 ✓

You're right, the idea that multiple epochs can't possibly help is one of the weakest links in the post. Sometime soon I hope to edit the post with a correction / expansion of that discussion, but I need to collect my thoughts more first -- I'm kinda confused by this too.

After thinking more about it, I agree that the repeated-data papers don't provide much evidence that multiple epochs are harmful.

For example, although the Anthropic repeated-data paper *does* consider cases where a non-small fraction of total training tokens are repeated more than once. In their most extreme case,

- half of the training tokens are never repeated during training, and
- the other half of training tokens are some (smaller) portion of the original dataset, repeated 2 or more times

But this effectively lowers the total size of the model's training dataset -- the number of training tokens is held constant (100B), so the repeated copies are taking up space that would otherwise be used for fresh data. For example, if the repeated tokens are repeated 2 times, then we are only using 3/4 of the data we could be (we select 1/2 for the unrepeated part, and then select 1/4 and repeat it twice for ... (read more)

[-] **ErickBall** 3y ▼ 17 ▲ ✕ 11 ✓

Thanks, that's interesting... the odd thing about using a single epoch, or even two epochs, is that you're treating the data points differently. To extract as much knowledge as possible from each data point (to approach L(D)), there should be some optimal combination of pre-training and learning rate. The very first step, starting from random weights, presumably can't extract high level knowledge very well because the model is still trying to learn low level trends like word frequency. So if the first batch has valuable high level patterns and you never revisit it, it's effectively leaving data on the table. Maybe with a large enough model (or a large enough batch size?) this effect isn't too bad though.

7 **Tao Lin** 3y This paper is very unrepresentative - it seems to test 1 vs 64-1,000,000 repeats ...

1 **Simon Lermen** 3y I can't access the wand link, maybe you have to change the access rules...

2 **nostalgebraist** 3y It should work now, sorry about that.

[-] **gwern** 3y ▼ 10 ▲ ✕ 3 ✓

> only if the model has more parameters than the dataset tokens and training for >10 epochs does overfitting kick in and scaling break down.

That sounds surprising. You are claiming that you observe the exact same loss, and downstream benchmarks, if you train a model on a dataset for 10 epochs as you do training on 10x more data for 1 epoch?

I would have expected some substantial degradation in efficiency such that the 10-epoch case was equivalent to training on 5x the data or something.

[-] **gwern** 3y ▼ 11 ▲ ✕ 1 ✓

Twitter points me to an instance of this with T5, Figure 6/Table 9: at the lowest tested level of 64 repeats, there is slight downstream benchmark harm but still a lot less than I would've guessed.

Not sure how strongly to take this: those benchmarks are weak, not very comprehensive, and wouldn't turn up harm to interesting capabilities like few-shots or emergent ones like inner-monologues; but on the other hand, T5 is also a pretty strong model-family, was SOTA in several ways at the time & the family regularly used in cutting-edge work still, and so it's notable that it's harmed so little.

[-] **harsimony** 3y* ▼ 17 ▲ ✕ 0 ✓

Some other order-of-magnitude estimates on available data, assuming words roughly equal tokens:

Wikipedia: 4B English words, according to this page.

Library of Congress: from this footnote a assume there are at most 100 million books worth of text in the LoC and from this page assume that books are 100k words, giving 10T words at most.

Constant writing: I estimate that a typical person writes at most 1000 words per day, with maybe 100 million people writing this amount of English on the internet. Over the last 10 years, these writers would have produced 370T words.

Research papers: this page estimates ~4m papers are published each year, at 10k words per paper with 100 years of research this amounts to 4T words total.

So it looks like 10T words is an optimistic order-of-magnitude estimate of the total amount of data available.

I assume the importance of a large quantity of clean text data will lead to the construction of a text database of ~1T tokens and that this database (or models trained on it) will eventually be open-sourced.

From there, it seems like really digging in to the sources of irreducible error will be necessary for further scaling. I would guess that a small part of this is... (read more)

2 **Peter Hroššo** 3y I think the models are evaluated on inputs that fill their whole context win...

1 **harsimony** 3y Oh I didn't realize! Thanks for clarifying. Uncertainty about location probabl...

[-] **Alex_Altair** 3y ▼ 12 ▲ ✕ 6 ✓

I have some thoughts that are either confusions, or suggestions for things that should be differently emphasized in this post (which is overall great!).

The first is that, as far as I can tell, these scaling laws are all determined empirically, as in, they literally trained a bunch of models with different parameters and then fit a curve to the points. This is totally fine, that's how a lot of things are discovered, and the fits look good to me, but a lot of this post reads as thought the law is a Law. For example;

> At least in terms of loss, Chinchilla doesn't just beat Gopher. It beats *any model trained on Gopher's data, no matter how big.*

This is not literally true, because saying "any model" could include totally different architectures that obey nothing like the empirical curves in this paper.

I'm generally unclear on what the scope of the empirical discovery is. (I'm also not particularly knowledgeable about machine learning.) Do we have reason to think that it applies in domains outside text completion? Does it apply to models that don't use transformers? (Is that even a thing now?) Does it apply across all the other bazillion parameters that go into a particular model, lik... (read more)

9 **nostalgebraist** 3y The answer to each these questions is either "yes" or "tentatively, yes." B...

3 **Alex_Altair** 3y Thanks! This whole answer was understandable and clarifying for me.

[-] **ESRogs** 3y ▼ 12 ▲ ✕ 4 ✓

Can you get anywhere with synthetic data? What happens if you train a model on its own output?

❗ 1

[-] **Legionnaire** 3y ▼ 12 ▲ ✕ 6 ✓

We're not running out of data to train on, just text.

Why did I not need 1 Trillion language examples to speak (debatable) intelligently? I'd suspect the reason is a combination of inherited training examples from my ancestors, but more importantly, language output is only the surface layer.

In order for language models to get much better, I suspect they need to be training on more than just language. It's difficult to talk intelligently about complex subjects if you've only ever read about them. Especially if you have no eyes, ears, or any other sense data. The best language models are still missing crucial context/info which could be gained through video, audio, and robotic IO.

Combined with this post, this would also suggest our hardware can already train more parameters than we need to in order to get much more intelligent models, if we can get that data from non text sources.

[-] **Dirichlet-to-Neumann** 3y ▼ 9 ▲ ✕ 12 ✓

Interesting and thought provoking.

"It's hard to tell, but there is this ominous comment, in the section where they talk about PaLM vs. Chinchilla:". In the context of fears about AI alignment, I would say "hopeful" rather than "ominous" !

[-] **Raemon** 3y Ω 5 ▼ 8 ▲ ✕ 0 ✓

Something I'm unsure about (commenting from my mod-perspective but not making a mod pronouncement) is how LW should relate to posts that lay out ideas that may advance AI capabilities.

My current understanding is that all major AI labs have already figured out the chinchilla results on their own, but that younger or less in-the-loop AI orgs may have needed to run experiments that took a couple months of staff time. This post was one of the most-read posts on LW this month, and shared heavily around twitter. It's plausible to me that spreading these ar... (read more)

[-] **Kaj_Sotala** 3y Ω 8 ▼ 19 ▲ ✕ 12 ✓

> so that the people who end up reading it are at least more likely to be plugged into the LW ecosystem and are also going to get exposed to arguments about AI risk.

There's also the chance that if these posts are not gated, people who previously *weren't* plugged into the LW ecosystem but are interested in AI find LW through articles such as this one. And then eventually also start reading other articles here and become more interested in alignment concerns.

There's also a bit of a negative stereotype among some AI researchers as alignment people being theoretical philosophers doing their own thing and being entirely out of touch about what real AI is like. They might take alignment concerns a bit more seriously if they find it easy to actually find competent AI discussion on LW / Alignment Forum.

[-] **nostalgebraist** 3y Ω 4 ▼ 10 ▲ ✕ 5 ✓

> My current understanding is that all major AI labs have already figured out the chinchilla results on their own, but that younger or less in-the-loop AI orgs may have needed to run experiments that took a couple months of staff time. This post was one of the most-read posts on LW this month, and shared heavily around twitter. It's plausible to me that spreading these arguments plausibly speeds up AI timelines by 1-4 weeks on average.

What is the mechanism you're imagining for this speedup? What happens that would not have happened without this post?

Consider that

- The Chinchilla paper was released over four months ago, on 3/29/22.
- It did not take long for the paper to get noticed among people interested in ML scaling, including here on LW.
  - On 3/29, the same day it was released, the paper was linked on r/mlscaling.
  - On 3/31, I heard about it through the EleutherAI discord, and immediately made an LW linkpost°.
  - On 4/1, 1a3orn posted a more detailed explainer°.

I'm struggling to imagine a situation where a relevant AI org is doing Chinchilla-like scaling experiments, yet somehow has managed to miss this paper (or to ignore/misunderstand it) for 4+ months. The paper is not exactl... (read more)

3 **Lech Mazur** 3y I don't have a strong opinion on hiding nostalgebraist's post behind a login g...

3 **Raemon** 3y Yeah a few people have also brought up this concern recently. Will think abou...

2 **Chris_Leong** 2y This is very tricky. On one hand, this may actually Streisand effect these res...

[-] **RyanCarey** 3y Ω 4 ▼ 8 ▲ ✕ 3 ✓

It would be useful to have a more descriptive title, like "Chinchilla's implications for data bottlenecks" or something.

[-] **Rodrigo Heck** 3y ▼ 8 ▲ ✕ 0 ✓

A possible avenue to explore is to expand these models to multilingual data. There are perhaps a lot of high quality text uniquely available in other languages (news, blogs, etc.). Anyways, IMO this effort should probably be directed less on acquiring the largest amount of data and more on acquiring high quality data. Chinchilla's scaling law doesn't include quality as a distinctive property, but we have reasons to believe that more challenging text are much more informative and can compensate low data environments.

[-] **Tom Lieberum** 3y ▼ 48 ▲ ✕ 11 ✓

I'd like to propose not talking publicly about ways to "fix" this issue. Insofar these results spell trouble for scaling up LLMs, this is a *good thing!*
Infohazard (meta-)discussions are thorny by their very nature and I don't want to discourage discussions around these results in general, e.g. how to interpret them or whether the analysis has merits.

[-] **nostalgebraist** 3y ▼ 74 ▲ ✕ 38 ✓

I disagree, but I'm not sure how relevant my opinion is, since I'm far less worried about "AGI ruin" to begin with than the median LWer. That said, here's my thinking:

First, there's no universally agreed-upon line between "discussing whether the analysis has merits" and "giving the capabilities people free ideas." Where a person draws this line depends on how obvious they think the ideas are, or how obvious they think they will be to the capabilities people.

Second, there are costs to not talking about things. It's useful for alignment research to have a correct sense of where capabilities research is headed, and where it isn't headed. If alignment researchers talk more to one another than to "capabilities people" (true IME), and they practice self-censorship like this, they'll end up with some importantly wrong beliefs.

Also, and perhaps worse -- if alignment researchers never voice their own secret capabilities ideas in fora where "capabilities people" can hear, then they'll never receive feedback about these ideas from the people who know what it would be like to apply them in the real world. Alignment researchers may end up with private stockpiles of... (read more)

[-] **gwern** 3y* ▼ 63 ▲ ✕ 25 ✓

> People would ask things like "what would it cost (in compute spending) to train a 10T parameter Chinchilla?", which is a bizarre way to frame things if you grok what Chinchilla is.

That wasn't an alignment researcher, though (was it? I thought Tomás was just an interested commenter), and it's a reasonable question to ask when no one's run the numbers, and when you get an answer like 'well, it'd take something like >5000x more compute than PaLM', that's a lesson learned.

At least among the people I've talked to, it seems reasonably well understood that Chinchilla had major implications, meant an immediate capabilities jump and cheaper deployment, and even more importantly meant parameter scaling was dead, and data and then compute were the bottleneck (which is also what I've said bluntly in my earlier comments), and this was why Chinchilla was more important than more splashy stuff like PaLM*. (One capability researcher, incidentally, wasn't revising plans but that's because he wasn't convinced Chinchilla was right in

the first place! AFAIK, there has been no dramatic followup to Chinchilla on part with GPT-3 following up Kaplan et al, and in fact, no one has replicated Chinchil... (read more)

[-] **Tomás B.** 3y ▼ 15 ▲ ✕ 4 ✓

> That wasn't an alignment researcher, though (was it? I thought Tomás was just an interested commenter)

Yep. Just an interested layman.

[-] **Lone Pine** 3y ▼ 12 ▲ ✕ 3 ✓

What are the public domain internet places where one can learn more about capabilities, or see discussions of capabilities frameworks? Here's what I'm aware of:

- LessWrong
- Twitter (but specifically who idk, I avoid Twitter for mental health reasons.)
- ArXiv (comp-sci)
- YouTube: MLST, Two Minute Papers, Yannic Kilcher, some conference talks
- A little bit on reddit (r/mlscaling, u/gwern)

All-in-all, there's not that much heavy discussion online. I've been told that these discussions really happen in-person, in the Bay Area and in DeepMind London offices. LessWrong actually ends up having the best discussion (*in the capabilities space.*)

(Since someone is likely to complain about seeking out more capabilities information, well yes it's risky, but I'm more in agreement with nostalgebraist that this level of discussion is probably harmless, and that it's better we keep an accurate and up-to-date understanding of the situation and technology.)

[-] **Hyperion** 3y ▼ 12 ▲ ✕ 5 ✓

Mostly Discord servers in my experience: EleutherAI is a big well known one but there are others with high concentrations of top ML researchers.

[-] **Leon Lang** 3y ▼ 11 ▲ ✕ 3 ✓

I upvoted since I think discussing what should or should not be discussed is important, but I tentatively disagree:

- It seems unlikely that comments on lesswrong speed up capabilities research since the thoughts are probably just a subset of what the scaling teams know, and lesswrong is likely not their highest signal information source anyway.
- Even from a safety perspective, it seems important to know which problems in capabilities research can be alleviated, since this will give a clearer picture of timelines.
- I think we should have strong reasons before discouraging topics of discussion since lesswrong is not only a place for instrumental rationality but also epistemic rationality -- maybe even more so.

That said, lesswrong is de facto one of the best places to discuss AI safety since the alignment forum is invite-only. thus, it seems that there should be some discussion around which tradeoffs to make on LW between "figuring out what's true" and "not spreading info hazards".

6 **Rob Bensinger** 3y I disagree with the reasoning in this reply to Tom (and in nostalgebrais...

6 **Leon Lang** 3y Thanks for your answer! This seems correct, though it's still valuable to fl...

4 **cata** 3y I think that argument is good if you expand out its reasoning. The reason we ha...

4 **Tom Lieberum** 3y Thanks for your reply! I think I basically agree with all of your points. I ...

[-] **Marius Hobbhahn** 3y ▼ 28 ▲ ✕ 12 ✓

My tentative heuristic for whether you should publish a post that is potentially infohazardy is "Has company-X-who-cares-mostly-about-capabilities likely thought about this already?". It's obviously non-trivial to answer that question but I'm pretty sure most companies who build LLMs have looked at Chinchilla and come to similar conclusions as this post. In case you're unsure, write up the post in a google doc and ask someone who has thought more about infohazards whether they would publish it or not.

Also, I think Leon underestimates how fast a post can spread even if it is just intended for an alignment audience on LW.

[-] **deepthoughtlife** 3y ▼ 7 ▲ ✕ 2 ✓

It would be quite easy to automatically generate all of the math and logic you could ever want for these models. Far more than you could possibly ever want train it on (wouldn't want to make it a math only bot, probably.). I could easily program a computer to come up with effectively infinite correct math problems. There are quintillions of 64bit addition problems alone... (actually an immense underestimate. there are 18.4 quintillion 64bit numbers alone). Subtraction, multiplication, division, algebra, trig, calculus, statistics, etc; AND, OR, NOT, XOR, N... (read more)

4 **Houshalter** 3y The Pile includes 7GB of math problems generated by deepmind basically as ...

2 **deepthoughtlife** 3y I am unsurprised it includes them, since it is an obvious thing. 7GB so...

3 **Houshalter** 3y Human beings can not do most math without pencil and paper and a lot ...

2 **deepthoughtlife** 3y I literally noted that GPT-J, which uses said 7GB of math (assuming ...

1 **Noosphere89** 3y The basic problem of arithmetic is this: You can't be informal in math,...

2 **deepthoughtlife** 3y You kind of can be informal though? Suppose, 5x - 2 = 3b +9, thu...

[-] **mgalle** 3y ▼ 6 ▲ ✕ 3 ✓

I know of two independently developed LLM in two languages where the conclusions of the developers is that "we run out of data in our language". One of them is trying to scale by going multilingual.

Where to look next? There is lots of untapped data in speech (radio shows, youtube, etc): that amount could make a difference in my opinion.

[-] **Lech Mazur** 3y ▼ 5 ▲ ✕ 1 ✓

This paper came out recently: https://arxiv.org/abs/2207.14502 . It shows a way to work around the lack of sufficient training data for generating computer programs by "generating synthetic programming puzzles and solutions, verified for correctness by a Python interpreter." We can think of analogous generation for data-limited general LLMs and there are some possibilities.

[-] **LGS** 3y ▼ 5 ▲ ✕ 0 ✓

Great post.

I have a question. Suppose we want to create a decent language model which is a small as possible -- small enough to run on a cell phone, say. We could try to compensate for this by scaling data to infinity. Now, we may run out of data, but if we do, we can generate more data artificially using a much larger LM. For example, consider training something BERT-sized using artificial data generated by PaLM (assume we have a very high compute budget in the training phase).

How well should we expect this to perform? If we plug into the above, it seems ... (read more)

4 **Lech Mazur** 3y You're describing a data augmentation variant of the teacher-student knowl...

[-] **Zvi** 3y Ω 3 ▼ 5 ▲ ✕ 0 ✓

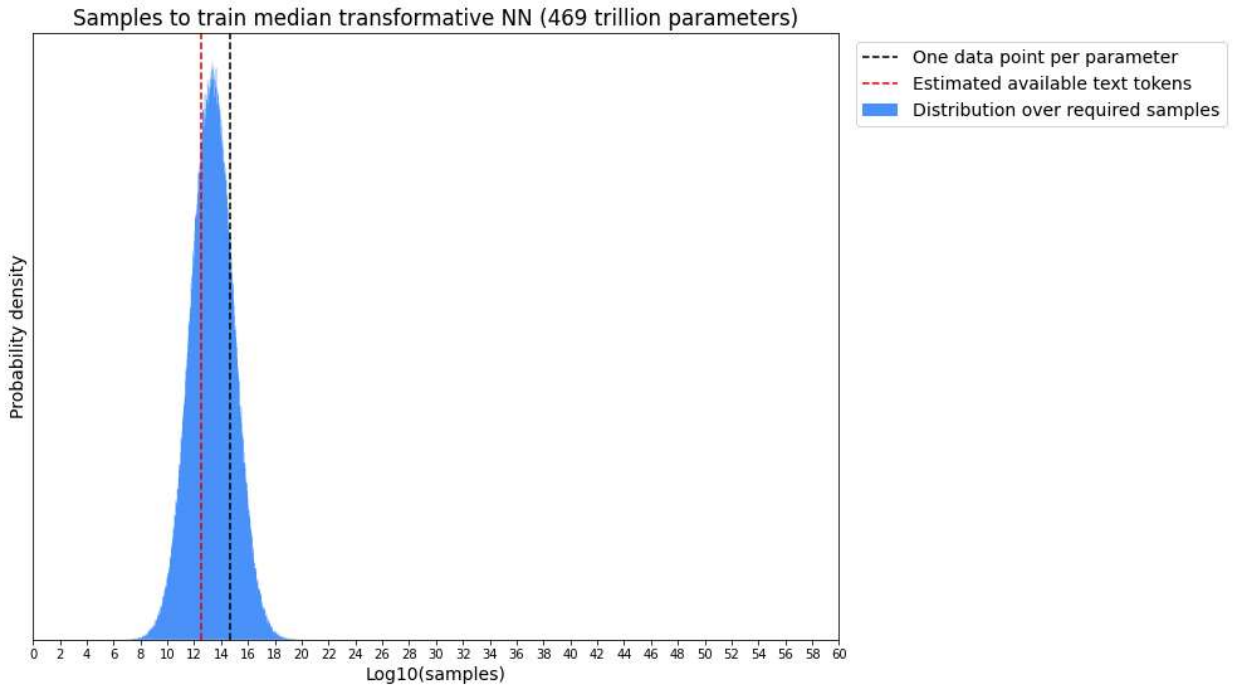
Scott Alexander asked things related to this, but still seems worth being more explicit about what this perfect 1.69 loss model would be like in practice if we got there?

6 **nostalgebraist** 3y The correct answer is the annoyingly trivial one: "it would be the best po...

8 **Vanessa Kosoy** 3y Transformers a Turing complete, so "model of this type" is not much of ...

[-] **aog** 3y ▼ 5 ▲ ✕ 0 ✓

It's worth noting that Ajeya's BioAnchors report estimates that TAI will require a median of 22T data points, nearly an order of magnitude more than the available text tokens as estimated here. See here ° for more.



[-] **traviswfisher@gmail.com** 3y ▼ 5 ▲ ✕ 0 ✓

Another interesting corpus (though problematic for legal reasons) would be sci-hub. Quick googling gives estimates of around 50 million research articles; the average research article runs around 4000 words, and sci-hub is estimated to contain about 69% of all research articles published in peer-reviewed journals. That would put sci-hub at about 50 million * 4000 = 200B tokens and the whole scientific journal literature at an estimated 290B tokens.

[-] **maxnadeau** 3y ▼ 5 ▲ ✕ 0 ✓

Confusion:

You write "Only PaLM looks better than Chinchilla here, mostly because it trained on 780B tokens instead of 300B or fewer, plus a small (!) boost from its larger size."

But earlier you write:

"**Chinchilla** is a model with the same training compute cost as Gopher, allocated more evenly between the two terms in the equation.

It's 70B params, trained on 1.4T tokens of data"

300B vs. 1.4T. Is this an error?

[-] **nostalgebraist** 3y ▼ 10 ▲ ✕ 1 ✓

Hmm, yeah, I phrased that point really badly. I'll go back and rewrite it.

A clearer version of the sentence might read:

> "Only PaLM is **remotely close to** Chinchilla here, mostly because it trained on a larger number of tokens than the other non-Chinchilla models, plus a small (!) boost from its larger size."

For instance, if you look at the loss improvement from Gopher to PaLM, 85% of it comes from the increase in data alone, and only 15% from the increase in model size. This is what I meant when I said that PaLM only got a "small" boost from its larger size.

EDIT: rewrote and expanded this part of the post.

3 **Buck** 3y I think that in that first sentence, OP is comparing PaLM to other large LMs rather t...

[-] **__nmca__** 3y ▼ 5 ▲ ✕ 0 ✓

Great post. The reason "no one was focused on data" was an error in the original OAI scaling laws paper, which was discovered by Hoffman et Al.

[-] **anon135711** 3y ▼ 4 ▲ ✕ 0 ✓

Super interesting post. I'm a bit late to the party, but I work in the space and the obvious reason nobody will say specifically what data they used is that their lawyers won't let them. I've literally had conversations with product counsel about how specific we can be in a paper / blog post about our data sources.

Copyright and privacy law is approximately 3 centuries behind the state of technology, and there are not clear rules about what data you're allowed to use for what. Someone like Google or Microsoft won't just say "we scraped Quo... (read more)

[-] **cubefox** 3y ▼ 4 ▲ ✕ 2 ✓

A comment from hacker news on this piece:

> The reason that language models require large amounts of data is because they lack grounding. When humans write a sentence about.. let's say "fire", we can relate that word to visual, auditory and kinesthetic experiences built from a coherent world model. Without this world model the LM needs a lot of examples, essentially it has to remember all the different contexts in which the word "fire" appears and figure out when it's appropriate to use this word in a sentence [...]

In other words, language models need so ... (read more)

[-] **metachirality** 2y ▼ 3 ▲ ✕ 2 ✓ *Review for 2022 Review*

Probably not the most important thing ever, but this is *really* pleasing to look at, from the layout to the helpful pictures, which makes it an absolute joy to read.

Also pretty good at explaining Chinchilla scaling too I guess.

[-] [anonymous] 3y ▼ 3 ▲ ✕ 0 ✓

Judging from this, might privacy regulations be one of the best ways to slow down AI development? Privacy is a widely accepted mainstream issue, so it should be a lot easier to advocate for. I think it

would be a lot easier for regular people to understand and get behind privacy regulation than DL regulation. On the other hand, it's not neglected and therefore less important on the margin.

-1 **The Hype Doesn't Help** 3y Why do you want regular people who aren't qualified to get inv...

1 **[anonymous]** 3y To give a short, very bad, but sort-of meaninfgful summary of my ideas: ...

[-] **tickybob** 3y ▼ 3 ▲ ✕ 0 ✓

There is an old (2013) paper from Google here that mentions training an ngram model on 1.3T tokens: (*"Our second-level distributed language model uses word 4-grams. The English model is trained on a 1.3 × 10^12 token training set").* An even earlier 2006 blog post here also references a 1T word corpus.

This number is 2x as big as MassiveWeb, more than a decade old, and not necessarily the whole web even back then. So I would be quite surprised if the MassiveWeb 506B token number represents a limit of what's available on the web. My guess would be that there'... (read more)

[-] **p.b.** 3y ▼ 3 ▲ ✕ 0 ✓

Some more questions:

> Meanwhile, the resulting model would not be nearly as big as PaLM. The optimal compute law actually puts it at 63B params.

How come PaLM_opt is smaller than Chinchilla? Isn't Chinchilla supposed to be Gopher_opt?

> Insofar as we trust our equation, this *entire* line of research -- which includes GPT-3, LaMDA, Gopher, Jurassic, and MT-NLG -- could *never* have beaten Chinchilla, no matter how big the models got[6].

These models where trained differently, which is why they had different scaling laws. Can we suppose that the new scalin... (read more)

7 **nostalgebraist** 3y See the footnote attached to that sentence. Great question, with a compl...

3 **p.b.** 3y Is the difference mostly the learning rate schedule? I read it was also AdamW and ...

6 **gwern** 3y (That is, the 'W' in AdamW stands for 'weight decay', that is, a lasso-like regula...

4 **CRG** 3y WD is not really about regularisation nowadays, so it's not surprising that it hel...

5 **gwern** 3y Yes, that's part of what I mean about regularization having weird effects an...

4 **Not Relevant** 3y Has this WD unimportance as regularization been written about som...

[-] **Aiyen** 3y ▼ 2 ▲ ✕ 0 ✓

On the MMLU benchmark, Chinchilla five-shot reported 67.6% accuracy; how does one convert this to loss or vice versa? More to the point, what loss would the human expert 89.8% correspond to? It would be very interesting to see how much compute that scaling law predicts would be necessary to produce human expert level losses with optimal data availability, or with as much data as is likely available to such a project.

[-] **Sushrut Karnik** 3y ▼ 1 ▲ ✕ 0 ✓

How many tokens would we have if we transcribed the audio of as many youtube videos as possible? (After a lot of filters I imagine)

[-] **Joey Yudelson** 3y ▼ 1 ▲ ✕ 0 ✓

Sorry if this is obvious, but where does the “irreducible” loss come from? Wouldn’t that also be a function of the data, or I guess the data’s predictability?

2 **nostalgebraist** 3y Yes, it's a function of the data, as well as the model architecture / training...

[-] **awlego** 3y ▼ 1 ▲ ✕ 0 ✓

I would expect the outcome of this to drive capabilities research more towards "learning to learn". Goal being to improve the amount of knowledge that is extracted from each observed piece of data.

[-] **Houshalter** 3y ▼ -1 ▲ ✕ -2 ✓

They fit a simplistic model where the two variables were independent and the contribution of each decays exponentially. This leads to the shocking conclusion that the two inputs are independent and decay exponentially...

I mean the model is probably fine for it's intended purpose; finding the rough optimal ratio of parameters and data for a given budget. It might mean that current models have suboptimal compute budgets. But it doesn't imply anything beyond that, like some hard limit to scaling given our data supply.

If the big tech companies really want to t... (read more)

3 **nostalgebraist** 3y What specific claims in the post do you disagree with? See this post for w...

2 **Houshalter** 2y I'm not sure what my exact thoughts were back then. I was/am at least ske...

Moderation Log

More from nostalgebraist

394	the void Ω	nostalgebraist	7mo	107
269	the case for CoT unfaithfulness is overstated Ω	nostalgebraist	1y	44
125	when will LLMs become human-level blogg...	nostalgebraist, DaemonicSigil	11mo	34

View more

Curated and popular this week

122	Why we are excited about confession!... ★	Boaz Barak, Gabriel Wu, Manas J...	5d	27
184	"The first two weeks are the hardest": my first digital declutte...	mingyuan	5d	9
153	Claude's new constitution	Zac Hatfield-Dodds, Drake Thomas	2d	30