Intuitions on language models

Jason Wei

OpenAI

Fundamental question. Why do large language models work so well?

Thing I've been thinking about recently: Manually inspecting data gives us clear intuitions about how the model works.

Looking at data = training your biological neural net.

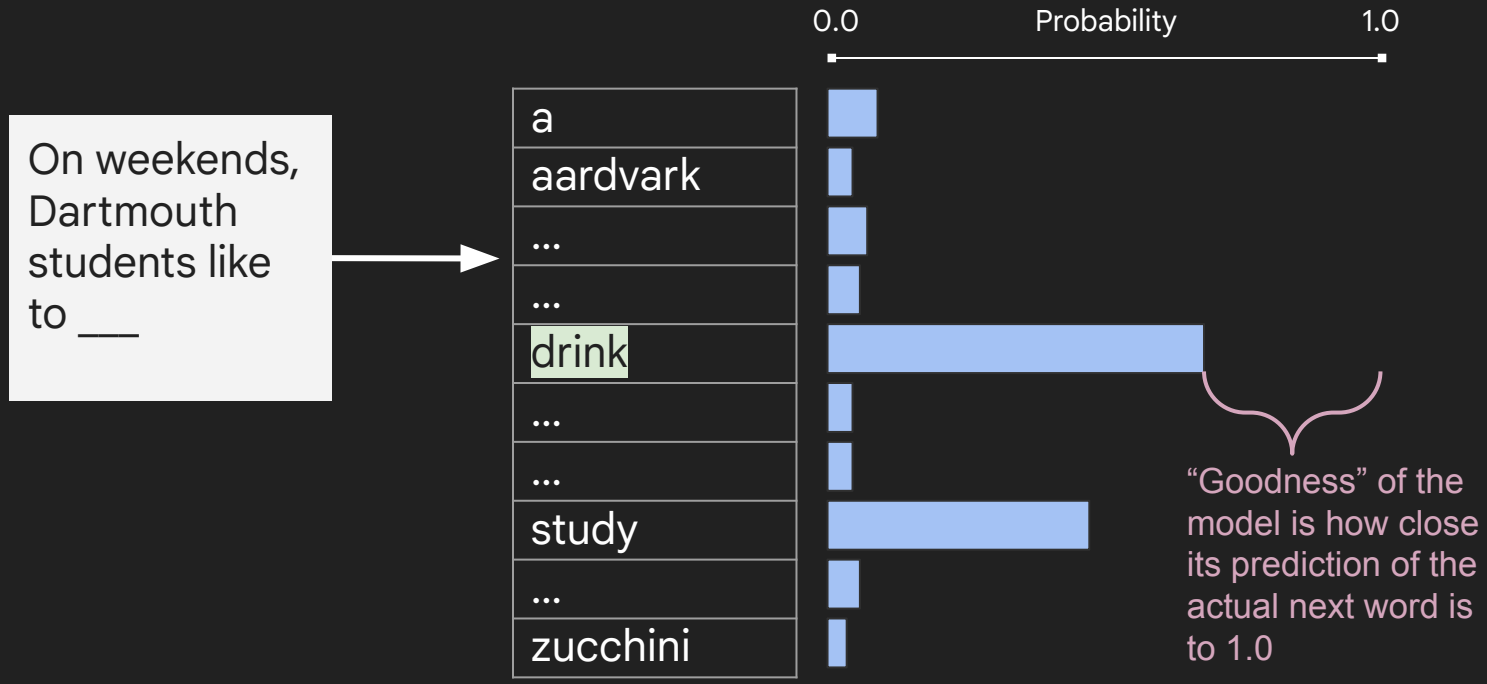
Your biological neural net makes many observations about the data after reading it.

These intuitions can be valuable.

(I once manually annotated an entire lung cancer image classification dataset. Several papers came out of intuitions from that process.)

Intuition 1.
Next-word prediction (on large data) is massively multi-task learning.

Language models are just next-word predictors



Example “tasks” from next-word prediction

Task	**Example sentence in pre-training that would teach that task**
Grammar	In my free time, I like to {**code**, **banana**}
Lexical semantics	I went to the store to buy papaya, dragon fruit, and {**durian**, **squirrel**}
World knowledge	The capital of Azerbaijan is {**Baku**, **London**}
Sentiment analysis	Movie review: I was engaged and on the edge of my seat the whole time. The movie was {**good**, **bad**}
Translation	The word for “pretty” in Spanish is {**bonita**, **hola**}
Spatial reasoning	Iroh went into the kitchen to make tea. Standing next to Iroh, Zuko pondered his destiny. Zuko left the {**kitchen**, **store**}
Math question	Arithmetic exam answer key: 3 + 8 + 4 = {**15**, **11**}

[millions more]

Extreme multi-task learning!

There are a lot of possible “tasks”, and they can be arbitrary

Input	Target	Task
Biden married Neilia	Hunter	world knowledge
Biden married Neilia Hunter	,	comma prediction
Biden married Neilia Hunter ,	a	grammar
Biden married Neilia Hunter , a	student	impossible?

https://en.wikipedia.org/wiki/Joe_Biden

Being a language model is not easy! A lot of arbitrary words to predict. Tasks aren’t weird and not clean.

Language models will learn what is in the data

ChatGPT (2022)

10 facts about jason wei

I'm sorry, but I don't have specific information about someone named Jason Wei. If you have any specific details or context about Jason Wei that you'd like to share, I can try to help based on that information.

ai researcher

As of my last update in January 2022, I don't have specific information about an AI researcher named Jason Wei. If Jason Wei is a more recent figure or someone who has become prominent after that time, I may not have information on him. If you have specific details or a context about his work or contributions to AI research, feel free to share, and I can help based on that information.

ChatGPT (2024)

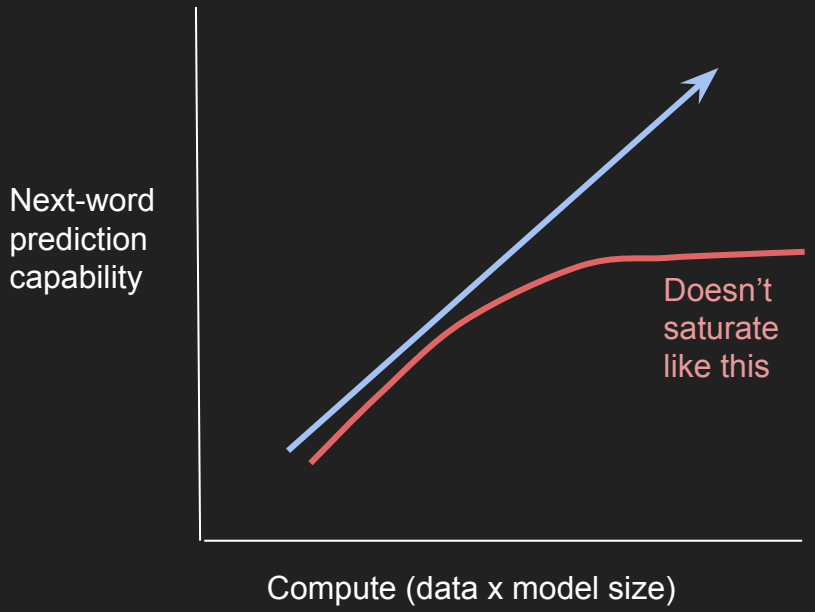
10 facts about jason wei

Jason Wei is a researcher and scientist known for his work in the field of artificial intelligence and natural language processing. Here are ten facts about him:

1. **Background**: Jason Wei holds a Ph.D. in Computer Science from Stanford University, where he conducted research in AI and NLP under the supervision of notable scholars in the field.
2. **Workplace**: Jason Wei has worked as a research scientist at Google Research, specifically in the Brain team, which focuses on advancing machine learning and AI technologies.
3. **Contributions to NLP**: He is known for his contributions to natural language processing (NLP), particularly in areas like few-shot learning, prompt engineering, and large language models.

Intuition 2.
Scaling language models (compute = model size * data) reliably improves loss.

Scaling predictably improves performance (“scaling laws”)



Kaplan et al., 2020:

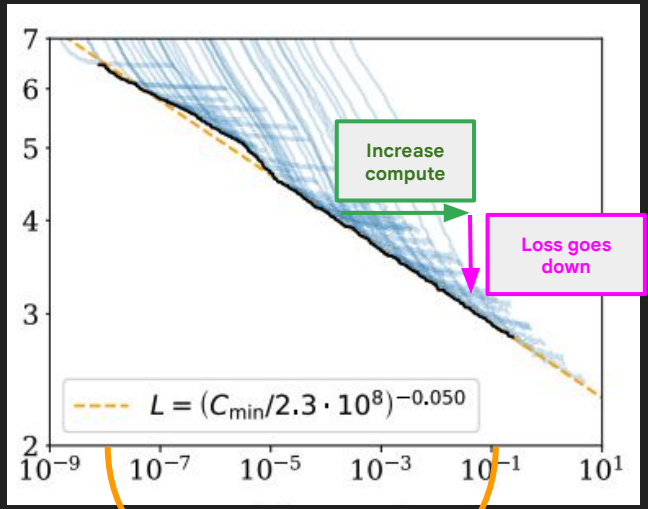
“Language modeling performance improves smoothly as we increase the model size, dataset size, and amount of compute for training.”

Jason’s rephrase: You should expect to get a better language model if you scale up compute.

Plot from the “scaling laws” paper

Scaling laws for neural language models. Kaplan et al., 2020.

Loss (lower is better, also log scale)



Compute (data x model size). Log scale!

Seven orders of magnitude

Scaling laws: certain metrics can be very predictable

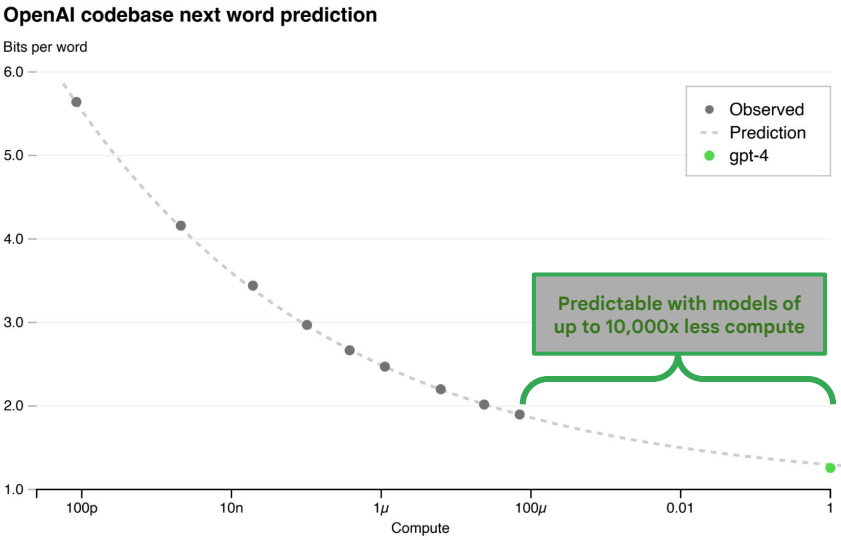


Figure 1. Performance of GPT-4 and smaller models. The metric is final loss on a dataset derived from our internal codebase. This is a convenient, large dataset of code tokens which is not contained in the training set. We chose to look at loss because it tends to be less noisy than other measures across different amounts of training compute. A power law fit to the smaller models (excluding GPT-4) is shown as the dotted line; this fit accurately predicts GPT-4's final loss. The x-axis is training compute normalized so that GPT-4 is 1.

GPT-4 technical report (2023).

Why does scaling work? Hard to answer, but here are some guesses

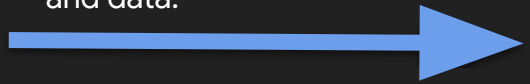
Small language model	**Large language model**
Memorization is costly *“Parameters are scarce, so I have to decide which facts are worth memorizing”*	More generous with memorizing tail knowledge *“I have a lot of parameters so I’ll just memorize all the facts, no worries”*
First-order correlations *“Wow, that token was hard. It was hard enough for me to even get it in the top-10 predictions. Just trying to predict reasonable stuff, I’m not destined for greatness.”*	Complex heuristics *“Wow, I got that one wrong. Maybe there’s something complicated going on here, let me try to figure it out. I want to be the GOAT.”*

Changes in the nature of AI work: scaling laws

5 years ago

- Many individual or small-scale projects.
- Bottom-up research culture.
- Run the code once; then submit to NeurIPS.

Technical paradigm shift: Training the best models requires scaling compute and data.



Now

- Teams have dozens of people.
- Everyone works together towards one focused goal.
- Tooling and infra matter a lot.

(In my experience, obviously there are exceptions.)

The general AI scaling law

~~Next-word prediction capability~~

Capabilities of AI

Time, money (compute, researchers) …

~~Compute~~

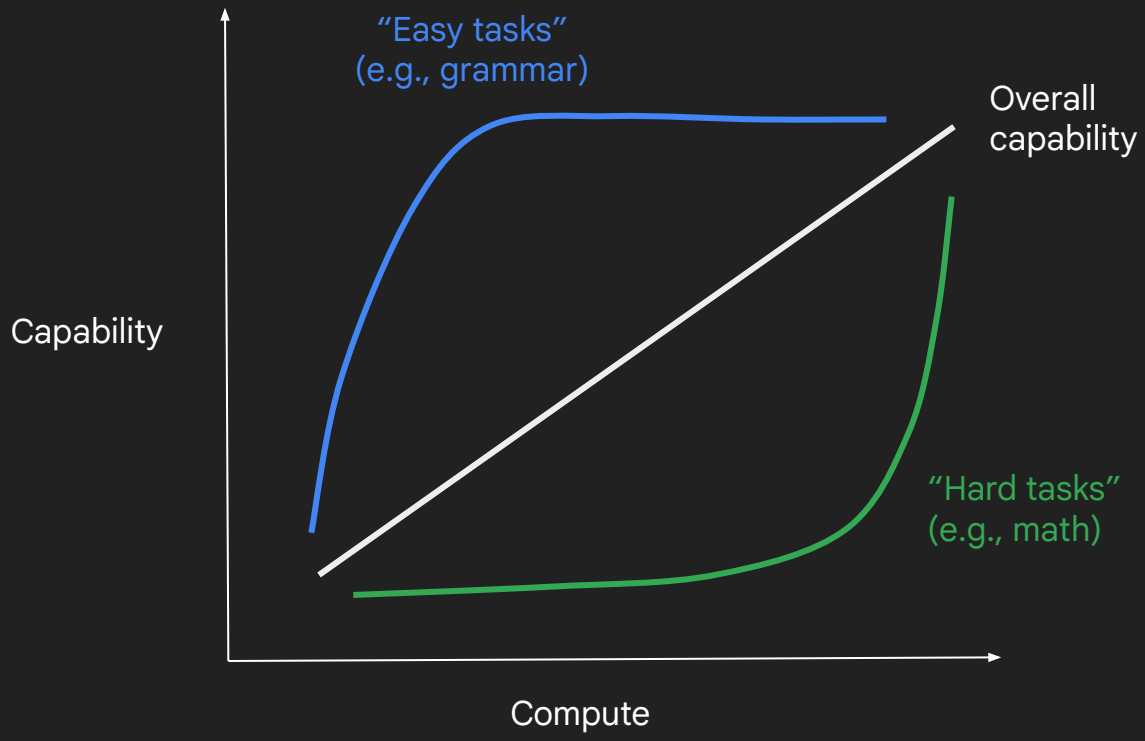
Intuition 3.
While overall accuracy scales smoothly, individual downstream tasks may improve in an emergent fashion.

Let’s take a closer look at accuracy. Consider:

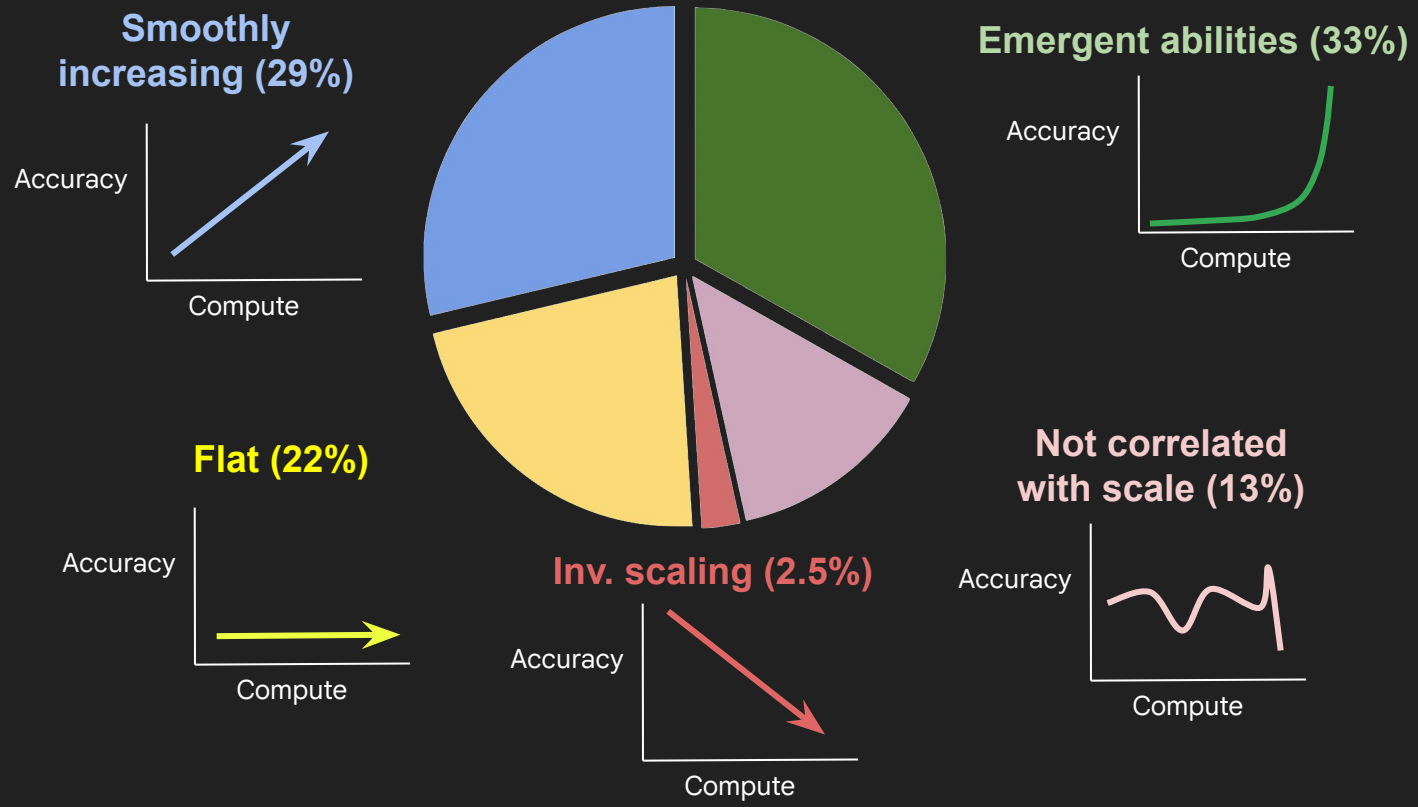
Overall accuracy = 1e-3 * accuracy_grammar +
1e-3 * accuracy_knowledge +
1e-6 * accuracy_sentiment_analysis +
...
1e-4 * accuracy_math_ability +
1e-6 * accuracy_spatial_reasoning
...

→ If accuracy goes from 70% to 80%, do all tasks get better uniformly?

Probably not.


"Easy tasks"
(e.g., grammar)
Overall
capability
Capability
"Hard tasks"
(e.g., math)
Compute

Real task distribution: 202 tasks in BIG-Bench

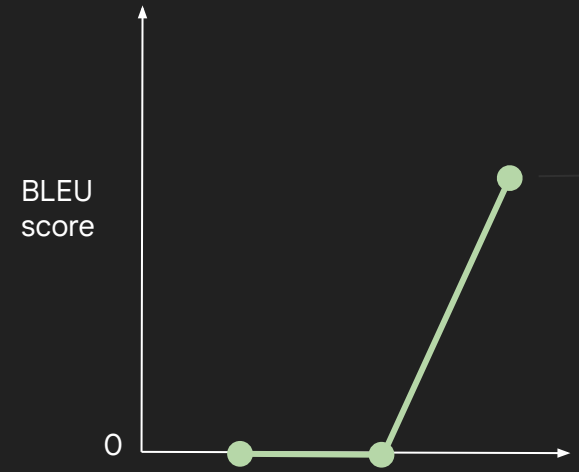


Emergence in prompting: example

Prompt

Input (English): I like to play soccer and tennis

Target (**Spanish**):



curie

Me gusta jugar al fútbol y al tenis ✓

ada

I like to play soccer and tennis ✗

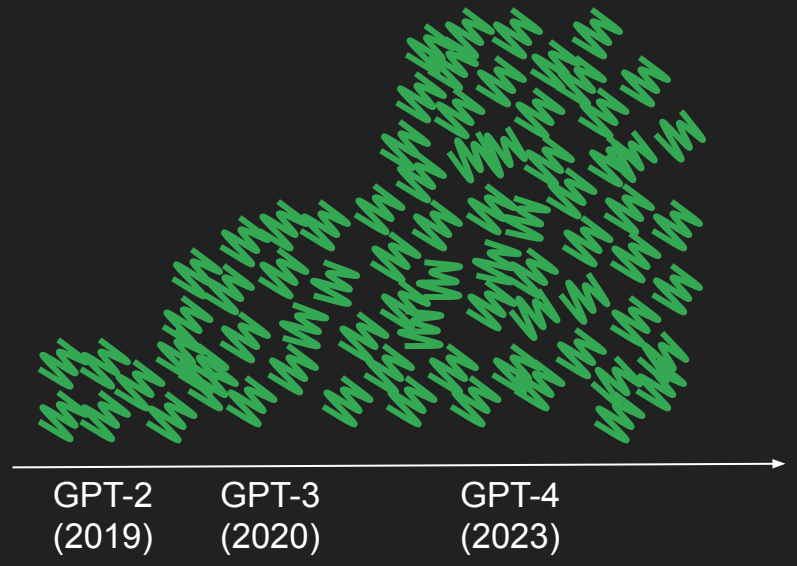
babbage

I like to play soccer and tennis ✗

Model “curie” suddenly figures out to translate and not repeat.

“Spectrum of possible tasks”

Write a novel
Scientific research
Hard math problems
…
Help debug code
Write a decent poem
Do basic math problems
Write a coherent essay
…
Translate a sentence
Write a summary
…
Give basic facts
Have correct grammar

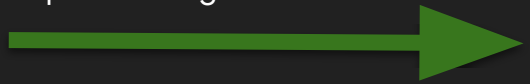


Changes in the nature of AI work: emergent abilities

5 years ago

- A few benchmarks for many years (CIFAR, ImageNet).
- Easy to rank models (top-1 acc, avg. SuperGLUE score).
- Task-specific architectures, data, and protocols.

Technical paradigm shift: A single model performs many tasks without the tasks being explicitly specified at pre-training.



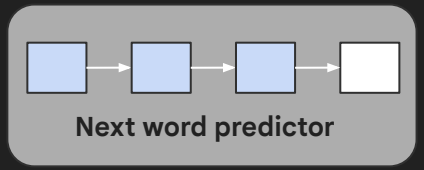
Now

- Need to create new benchmarks all the time.
- Hard to decide if one model is universally better.
- Create general technology; relatively easy to pivot.

Thanks.

X / Twitter: @_jasonwei

How to think about language model pre-training



How to think about language model pre-training

Millions of tasks!