Don’t teach. Incentivize.

MIT EI seminar

Hyung Won Chung

OpenAI

Non-goal: share specific technical knowledge and experimental results

Goal: share how I think with AI being a running example

Why?

We, the technical people, focus too much on problem solving itself

In my view, more attention should go to finding great problems to solve

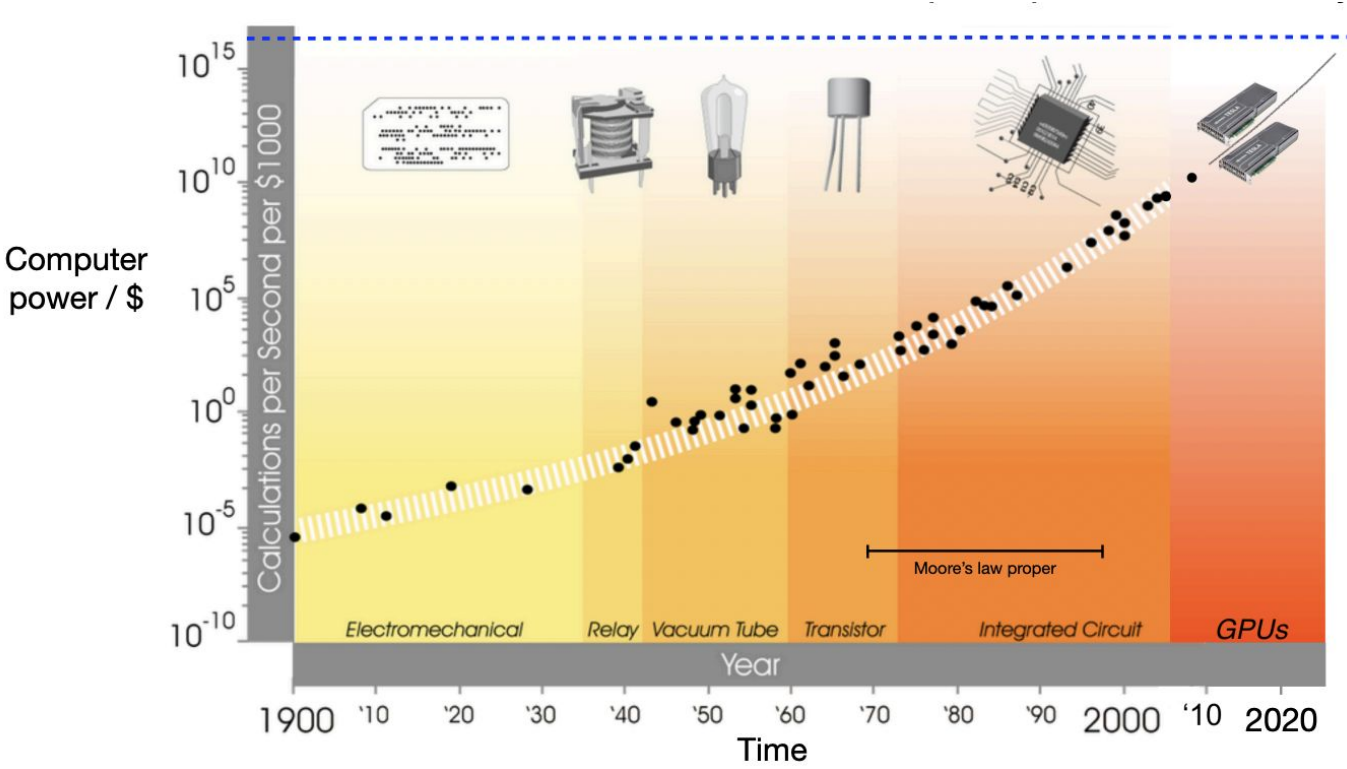
Great researchers are good at finding impactful problems. I think this ability comes from having the right perspective.

I hope this talk sparks interest in developing original perspectives, which in turn help finding better problems to solve

Outline

Build the scale-first perspective for AI research in general

Interpret Large Language Models with this perspective



Roughly, 10x more compute every 5 years

Figure from Rich Sutton's WAIC keynote

Hardware is exponentially progressing

Software and algorithms should catch up

We need more **scalable** methods that can better leverage computation

The job of AI researchers is to teach machines how to “think”

One (unfortunately common) approach

Teach the machines how *we think* we think

But we don’t know how we think at the neuron level

So we are teaching what we don’t fully understand in a limited language of mathematics

This approach poses structures to the problem, which can become the limitation when scaled up

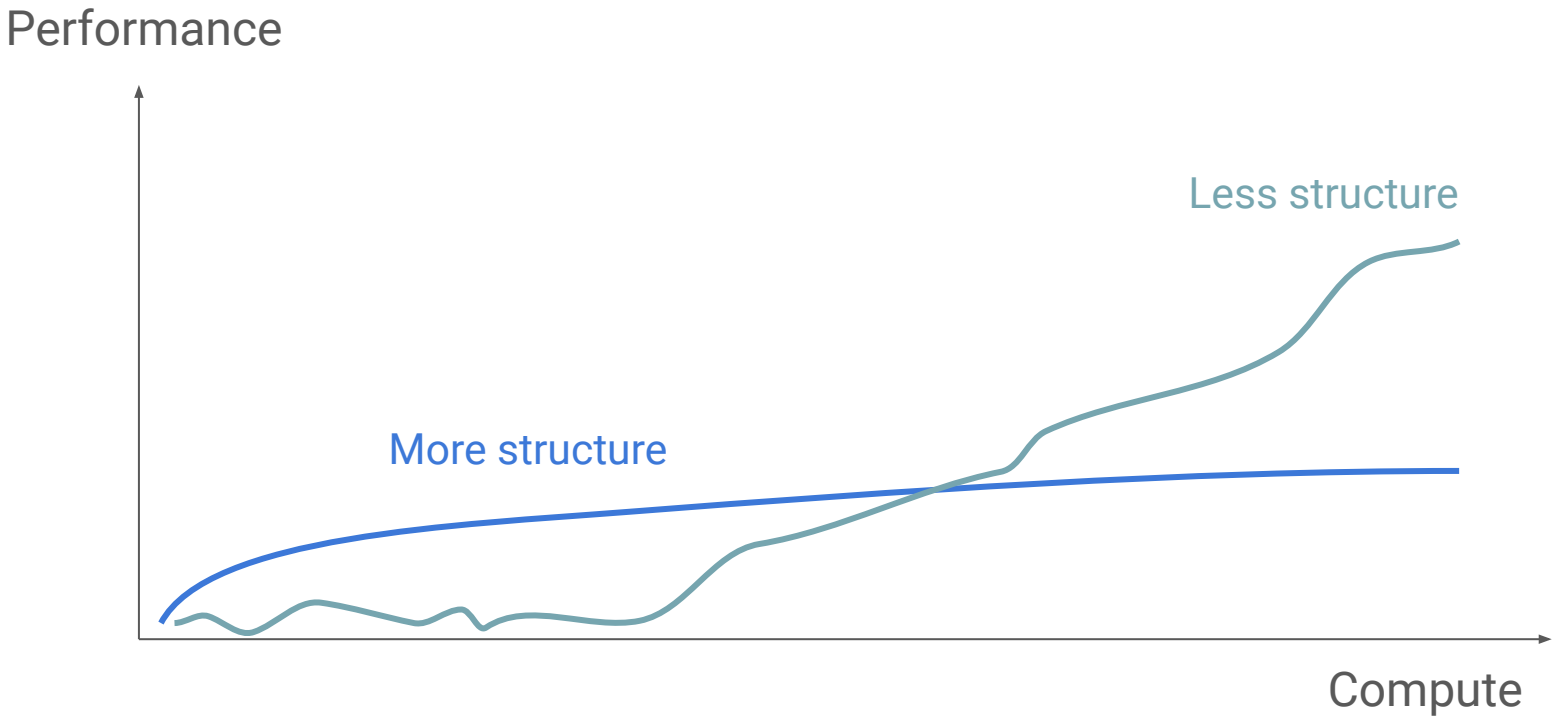
Bitter lesson

Progress of AI in the past 70 years boils down to

- Develop progressively more general methods with less structure
- Add more data and computation (i.e. scale up)

http://www.incompleteideas.net/IncIdeas/BitterLesson.html

The more structure imposed by humans, the less scalable the method is



Sobering observation

Clever structures posed by human researchers typically become the bottleneck when scaled up

What is good in the long run almost necessarily looks bad in the short term

Compute is getting cheaper faster than we are becoming better researchers

Give machines more degrees of freedom. Let them choose how they learn

Why are these observations not so obvious?

Researchers want to add modeling idea because that is academically more satisfying

Some people think “just scaling up” is not scientific or interesting

Ultimately what do we want to achieve with artificial intelligence?

We should focus on:

maximizing the value generated by AI while minimizing the downside

regardless of which academic discipline achieves the goal

HWC’s definition of scaling

Common definition: doing the same thing with more machines

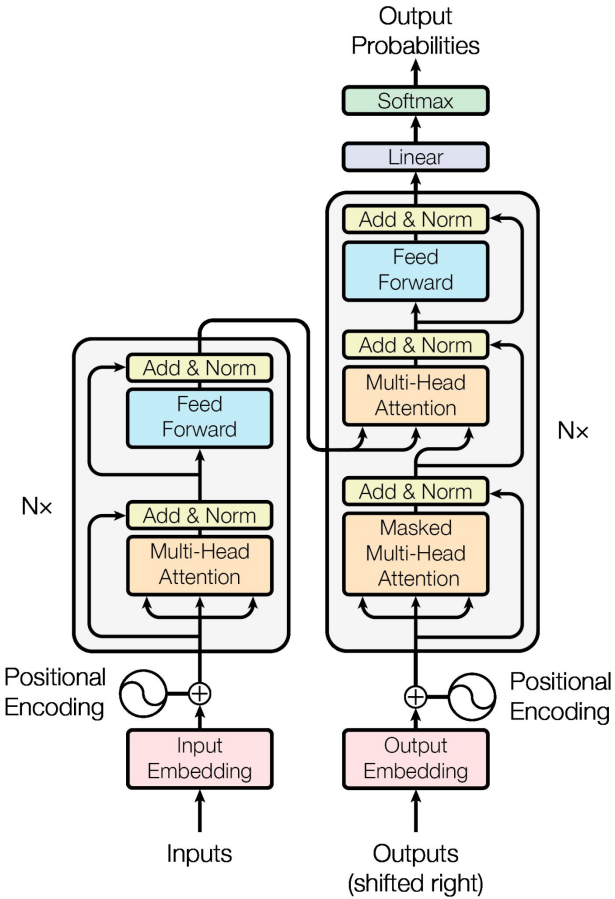
HWC's definition of scaling

Common definition: doing the same thing with more machines

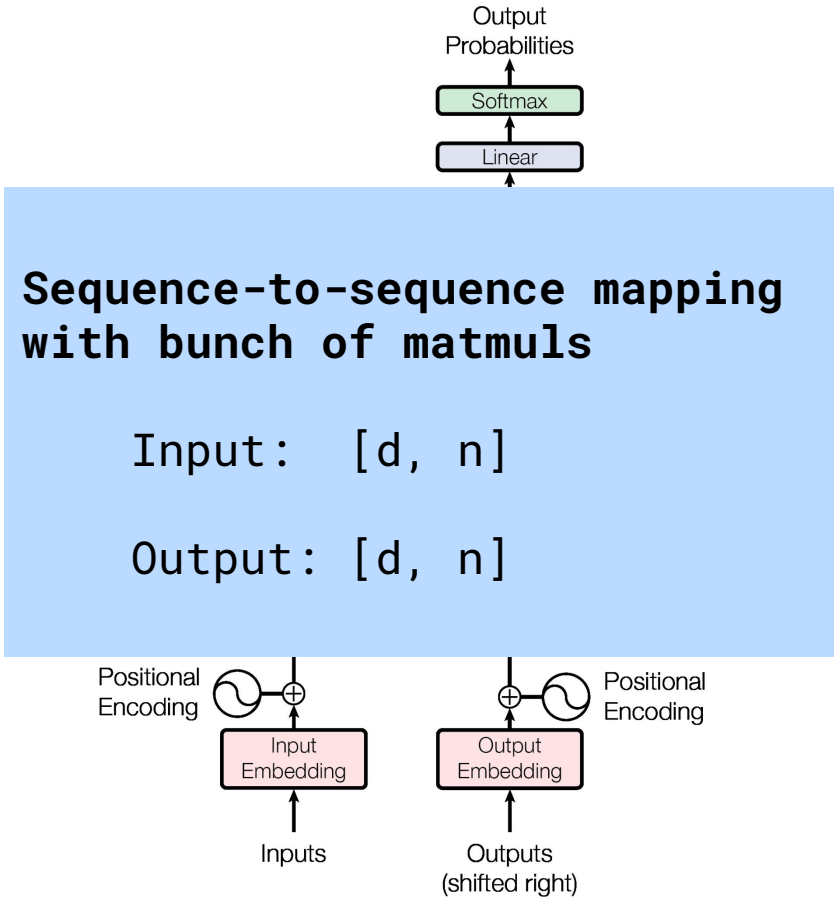
Scaling implicitly involves *identifying* the modeling assumption that bottlenecks further scaling and *replacing* it with a more scalable one

Large Language Models (LLMs)

All LLMs so far use Transformer architecture



Let's take a "functional" viewpoint on the Transformer



Process	Shape
“Many words don't map to one token: indivisible.”	[ ]

Process	Shape
“Many words don't map to one token: indivisible.”	[ ]
⬇ Tokenization	⬇
Many words don't map to one token: indivisible.	
[7085, 2456, 836, 470, 3975, 284, 530, 11241, 25, 773, 452, 12843, 13]	[n]

Process

Shape

“Many words don't map to one token: indivisible.”

[ ]

⬇ Tokenization

⬇

Many words don't map to one token: indivisible.

[7085, 2456, 836, 470, 3975, 284, 530, 11241, 25, 773, 452, 12843, 13]

[n]

⬇ Embedding

⬇

2.3	-3.2	8.3	5.4	2.1	3.9	-8.9	3.8	3.9	3.3
4.5	5.9	4.5	7.1	1.0	5.3	5.0	3.1	0.7	5.0
…	…	…	…	…	…	…	…	…	…
3.8	1.2	3.8	9.0	9.3	3.1	4.2	0.8	9.2	5.8

[d, n]

Process

Shape

“Many words don't map to one token: indivisible.”

[ ]

↓ Tokenization

Many words don't map to one token: indivisible.

[7085, 2456, 836, 470, 3975, 284, 530, 11241, 25, 773, 452, 12843, 13]

[n]

↓ Embedding

2.3	-3.2	8.3	5.4	2.1	3.9	-8.9	3.8	3.9	3.3
4.5	5.9	4.5	7.1	1.0	5.3	5.0	3.1	0.7	5.0
…	…	…	…	…	…	…	…	…	…
3.8	1.2	3.8	9.0	9.3	3.1	4.2	0.8	9.2	5.8

[d, n]

↓ N Transformer layers

3.2	-2.3	3.8	4.5	1.2	9.3	-9.8	8.3	9.3	3.3
5.4	9.5	5.4	1.7	0.1	3.5	0.5	1.3	7.0	0.5
…	…	…	…	…	…	…	…	…	…
8.3	2.1	8.3	0.9	3.9	1.3	2.4	8.0	2.9	8.5

[d, n]

Process	Shape
“Many words don't map to one token: indivisible.”	[ ]
⬇ Tokenization	⬇
Many words don't map to one token: indivisible.	
[7085, 2456, 836, 470, 3975, 284, 530, 11241, 25, 773, 452, 12843, 13]	[n]
⬇ Embedding	⬇
2.3 -3.2 8.3 5.4 2.1 3.9 -8.9 3.8 3.9 3.3 4.5 5.9 4.5 7.1 1.0 5.3 5.0 3.1 0.7 5.0 … … … … … … … … … … 3.8 1.2 3.8 9.0 9.3 3.1 4.2 0.8 9.2 5.8	[d, n]
⬇ N Transformer layers	⬇
3.2 -2.3 3.8 4.5 1.2 9.3 -9.8 8.3 9.3 3.3 5.4 9.5 5.4 1.7 0.1 3.5 0.5 1.3 7.0 0.5 … … … … … … … … … … 8.3 2.1 8.3 0.9 3.9 1.3 2.4 8.0 2.9 8.5	[d, n]
⬇ Loss function (predict next token given previous)	⬇
2.6	[ ]

Original sentence

```
Many words don't map to one token: indivisible.
```

Original sentence

```
Many words don't map to one token: indivisible.
```

Given “many”, predict the next token

```
Many
```

apple: 0.01
don: 0.001
…
intelligence: 0.00001
…
words: 0.02

Original sentence

```
Many words don't map to one token: indivisible.
```

Given “many”, predict the next token

```
Many
```

apple: 0.01
don: 0.001
…
intelligence: 0.00001
…
words: 0.02

Given “many words”, predict the next token

```
Many words
```

apple: 0.00003
don: 0.03
…
intelligence: 0.00001
…
words: 0.0000001

Original sentence

```
Many words don't map to one token: indivisible.
```

Given “many”, predict the next token

```
Many
```

apple: 0.01
don: 0.001
…
intelligence: 0.00001
…
words: 0.02

Given “many words”, predict the next token

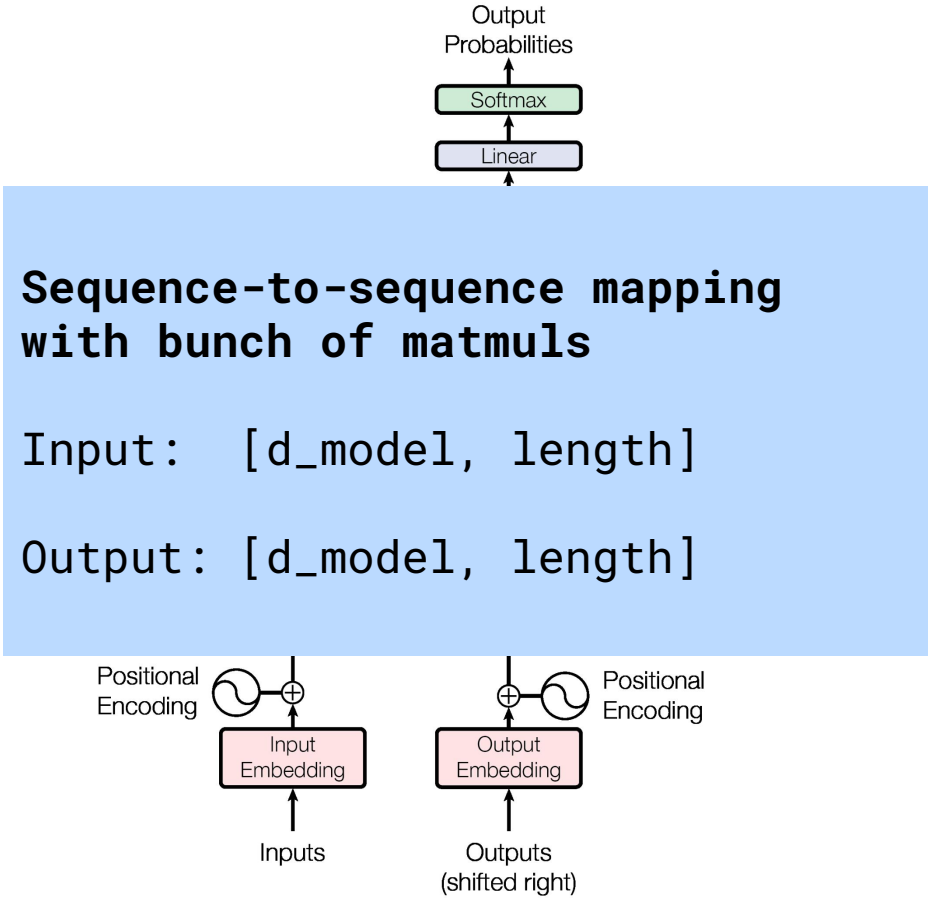
```
Many words
```

apple: 0.00003
don: 0.03
…
intelligence: 0.00001
…
words: 0.0000001

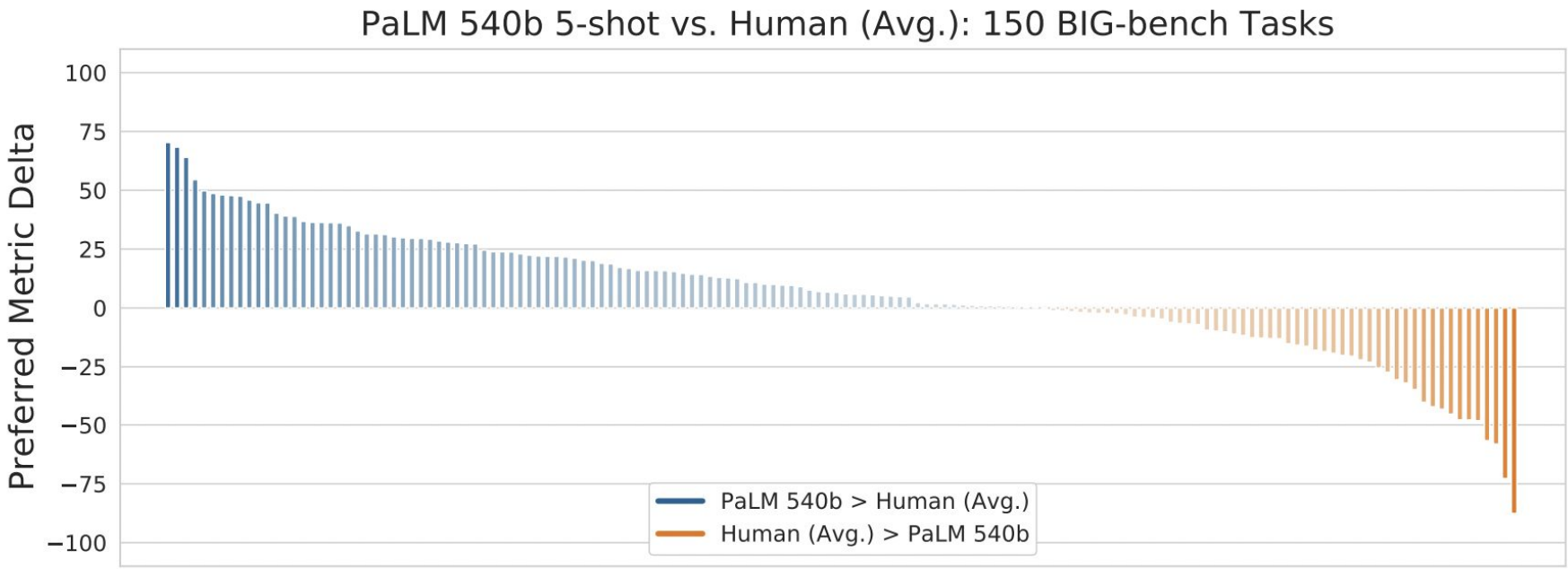
Probability of a sentence is a product of conditional probabilities. Maximize this.

Feed web-scale text data to Transformer

Web-scale text data



Somehow the model learns to perform many many tasks only trained with next-token prediction



Chowdhery et al (2022)

Some observations on the next-token prediction task

We don’t directly teach any linguistic concepts (e.g. verb, subject, whatever)

Simply by predicting next tokens over a large corpus, the model learns languages

Language is learned almost as a *by-product* of doing such task

The model can do some “reasoning” (e.g. math, code)

Next token prediction as a massive implicit multitask learning

Next token prediction as a massive implicit multitask learning

This terrible movie was really boring

Next token prediction as a massive implicit multitask learning

This terrible movie was really boring

After the earning call, the share price of Google went up by 5% from $1,000, ending in $1,050

Next token prediction as a massive implicit multitask learning

This terrible movie was really boring

After the earning call, the share price of Google went up by 5% from $1,000, ending in $1,050

인공지능 연구원들은 코딩을 잘
못합니다.

Next token prediction as a massive implicit multitask learning

This terrible movie was really boring

After the earning call, the share price of Google went up by 5% from $1,000, ending in $1,050

인공지능 연구원들은 코딩을 잘
못합니다.

The first law of Thermodynamics is often called conservation of energy

Next token prediction as a massive implicit multitask learning

This terrible movie was really boring

After the earning call, the share price of Google went up by 5% from $1,000, ending in $1,050

인공지능 연구원들은 코딩을 잘 못합니다.

The first law of Thermodynamics is often called conservation of energy

BILLIONS of sentences

TRILLIONS of task types

Massive multitask learning hypothesis

Beyond some scale, the easiest way to do well on the next token prediction is for the model to find a set of general skills that are applicable to many tasks.

For example, these skills include learning languages, understanding and reasoning.

Crucially we don’t directly teach any of these skills to the model. We weakly *incentivize* the model and the abilities *emerge*

Abilities that emerge are typically more general skill sets. In order for abilities to emerge, they should be incentivized as opposed to being directly taught

Weakly incentivizing the model requires a lot more compute, i.e. it is a more scalable teaching strategy

For a given dataset and an learning objective there is an explicit learning signal and a set of induced incentives

Next-token prediction with web-scale data

- explicit signal: predict next token
- induced incentive: understand languages and reasoning, etc

Example 2: Playing chess with {0, 1} reward at the end of the game

Explicit signal: win the game

Induced incentive: learn what moves are good

Example 3: Hallucinations

Reward structure for simple question answering scenario:

- 1 if the answer is correct and unhedged
- 0.5 if answer is correct but hedged
- 0 if the answer is “I don’t know”
- -2 if the answer is hedged but wrong
- -4 if the answer is unhedged and wrong

Explicit signal: answer the question correctly

Induced incentive: know what you don’t know

Adapted from John Schulman’s talk https://www.youtube.com/watch?v=hhiLw5Q_UFg

Loose analogy

Give a man a fish, and you feed him for a day.

Teach a man to fish, and you feed him for a lifetime.

Loose analogy

Give a man a fish, and you feed him for a day.

Teach a man to fish, and you feed him for a lifetime.

Teach him the taste of fish and make him hungry

Give a man a fish

Teach him how to fish

Teach him the taste of fish and make him hungry

Time required

Give a man a fish

Teach him how to fish

Teach him the taste of fish and make him hungry

humans

Time required

machines

Compute required

Small specialist models vs large generalist model

The belief that small specialist models can win on a narrow domain assumes that there exists tradeoffs between being a generalist and specialist

Specialist-generalist tradeoff doesn’t apply to machines

Such tradeoff is due to the fact that every human beings operate with the same time budget. Machines do not.

One model gets to enjoy a lot more compute than others

It is akin to someone having access to “Room of spirit and time” from Dragon ball; one year inside that room is a day outside

Importance of incentive structure is not a new. Why now?

No amount of bananas can incentivize monkeys to do mathematical reasoning

Threshold intelligence is necessary for the incentive structure to work for a given problem

I think we cross that threshold for many tasks

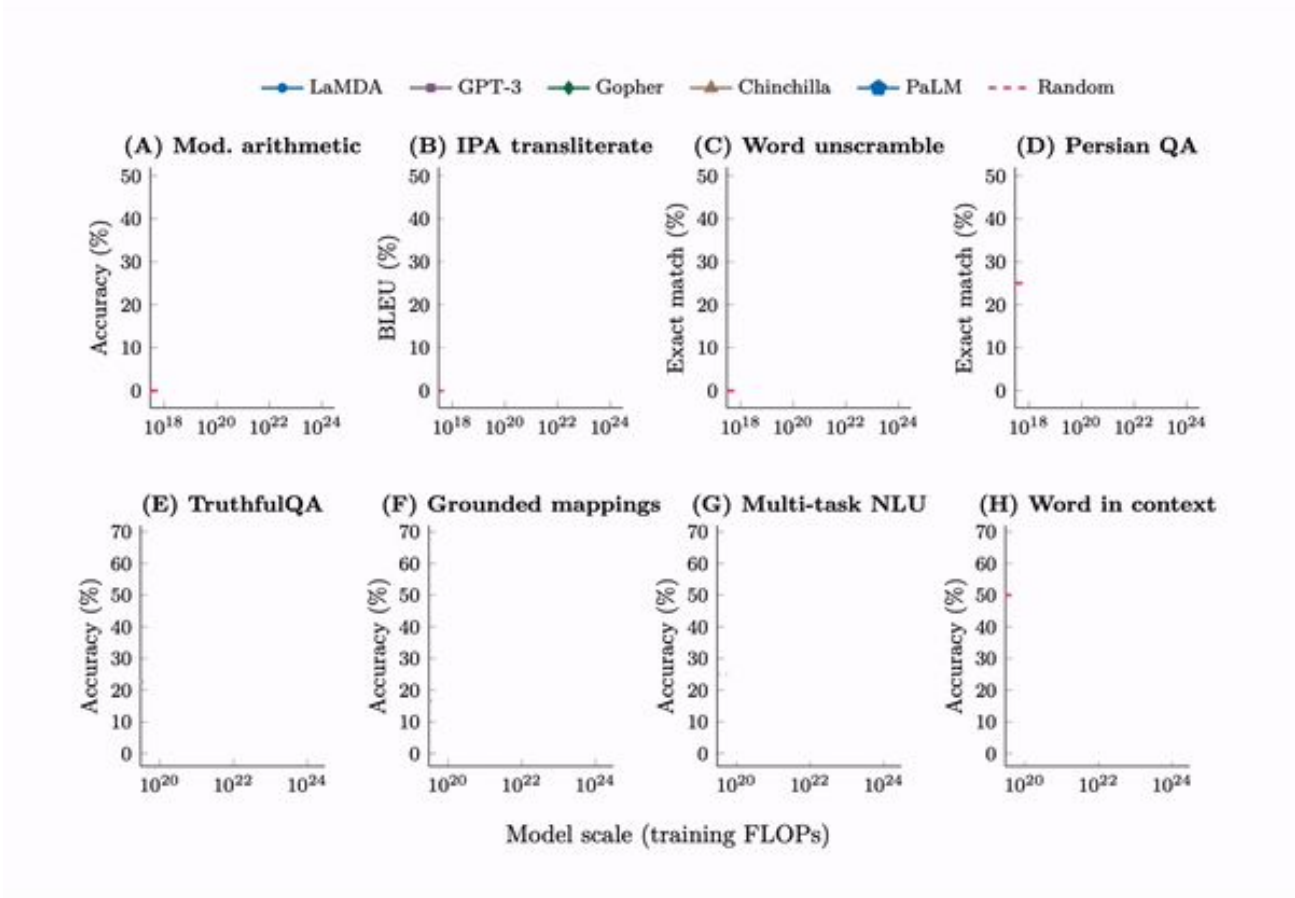
Whether the induced incentive structure works depends on the model size

What abilities emerge depends on the model size

If the model is too small, the model might just give up learning high-level skills such as reasoning. It relies on heuristics-based pattern recognition

Some abilities emerge with scale

Having the right perspective is crucial



[Emergent Abilities of Large Language Models](#)
Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph et al. (2022)

Perspective of “yet”

Perspective of “yet”

This idea doesn’t work

Perspective of “yet”

This idea doesn’t work  This idea doesn’t work *yet*

Why is the perspective of “yet” not so obvious?

We are used to operating in an environment where underlying axioms don’t change

You run an experiment for your new scientific idea. It doesn’t work now. You know that it will not work if you run 3 years later

For language models, the most capable model serves as an “axiom” for many research experiments run on top

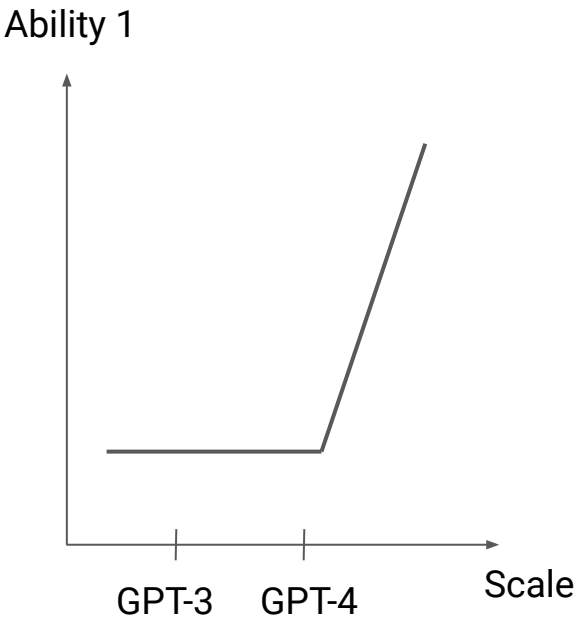
Need for constant unlearning

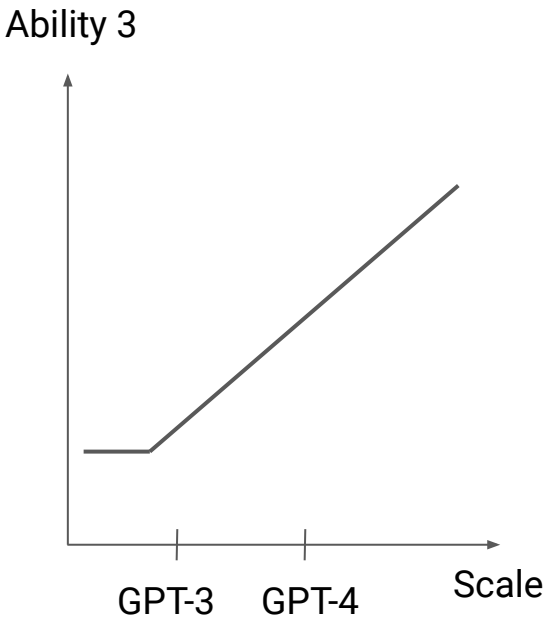
Many ideas get outdated and invalidated at larger scale

We need to constantly unlearn intuitions built on such invalidated ideas

With less to unlearn, newcomers can have advantages over more experienced ones. This is an interesting neutralizing force

Highly simplified view of emergent abilities





Closing

Compute cost is decreasing exponentially

AI researchers should harness this by designing scalable methods

Current generation of LLMs rely on next-token prediction, which can be thought of as weak incentive structure to learn general skills such as reasoning

More generally, we should incentivize models instead of directly teaching specific skills

Emergent abilities necessitate having the right perspective such as unlearning

Thank you!

Twitter: @hwchung27

Don’t teach. Incentivize.

MIT EI seminar

Hyung Won Chung

OpenAI