

# 2050: An AI Odyssey: Dark Matter of Intelligence

Yejín Choi

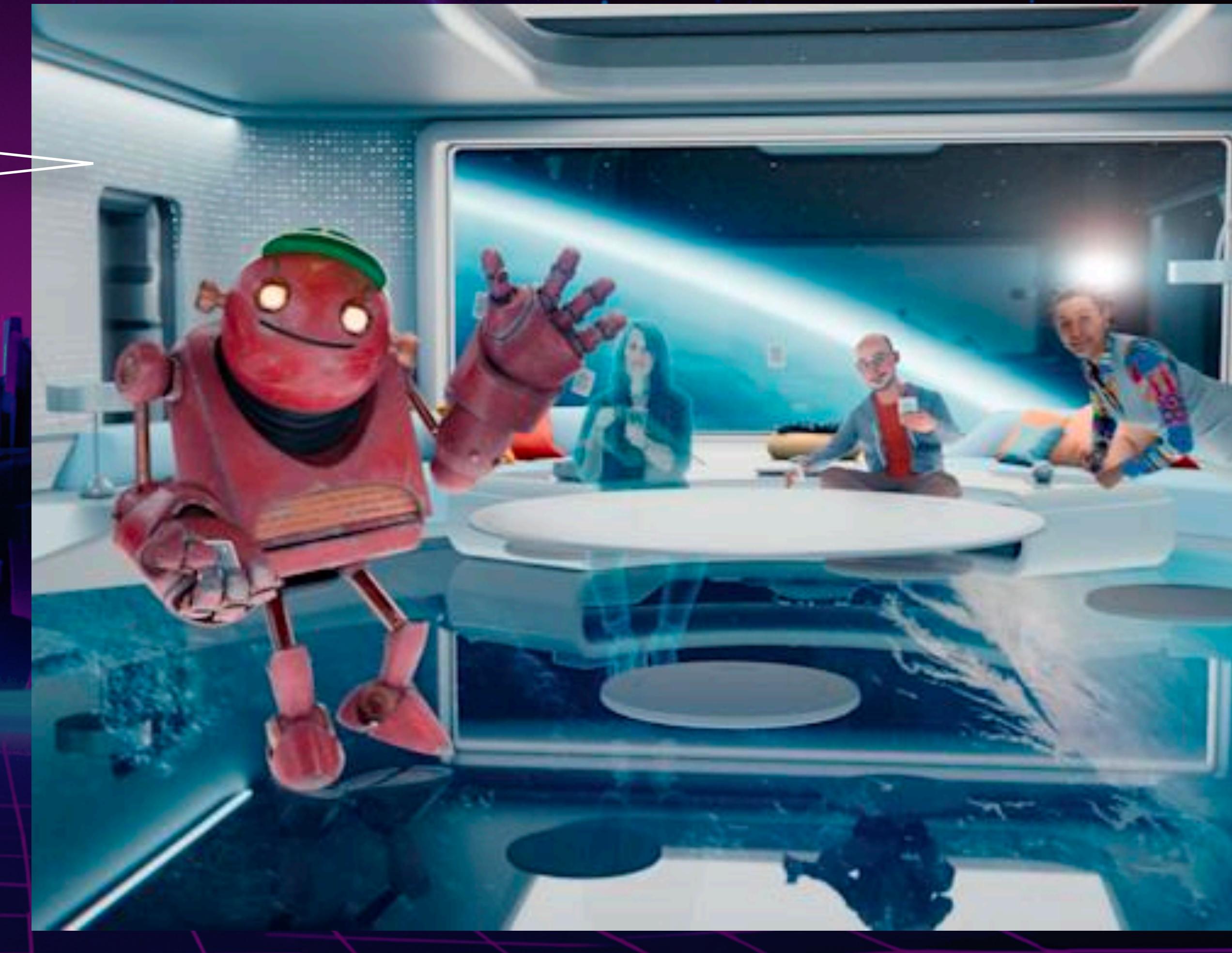
University of Washington & Ai2



what CVPR 2050 be like?

# What CVPR 2050 be like?

Venue: metaverse?



# What CVPR 2050 be like?

Venue: mars?



# What CVPR 2050 be like?

ChatGPT writes the paper

ChatGPT reviews the paper

ChatGPT rebuttal period

Diffusion generates slides

NeRF presents the talk

ChatGPT summarizes the talk?

Few-shot prompting &

Instruction tuning?

NeRF? Diffusion? Transformers?

Autonomous driving? cleaning?  
plumbing? babyseating?

LLMs (or LVMs?) as prior?

Scaling laws no more?



What

ChatGPT writes the

ChatGPT reviews the

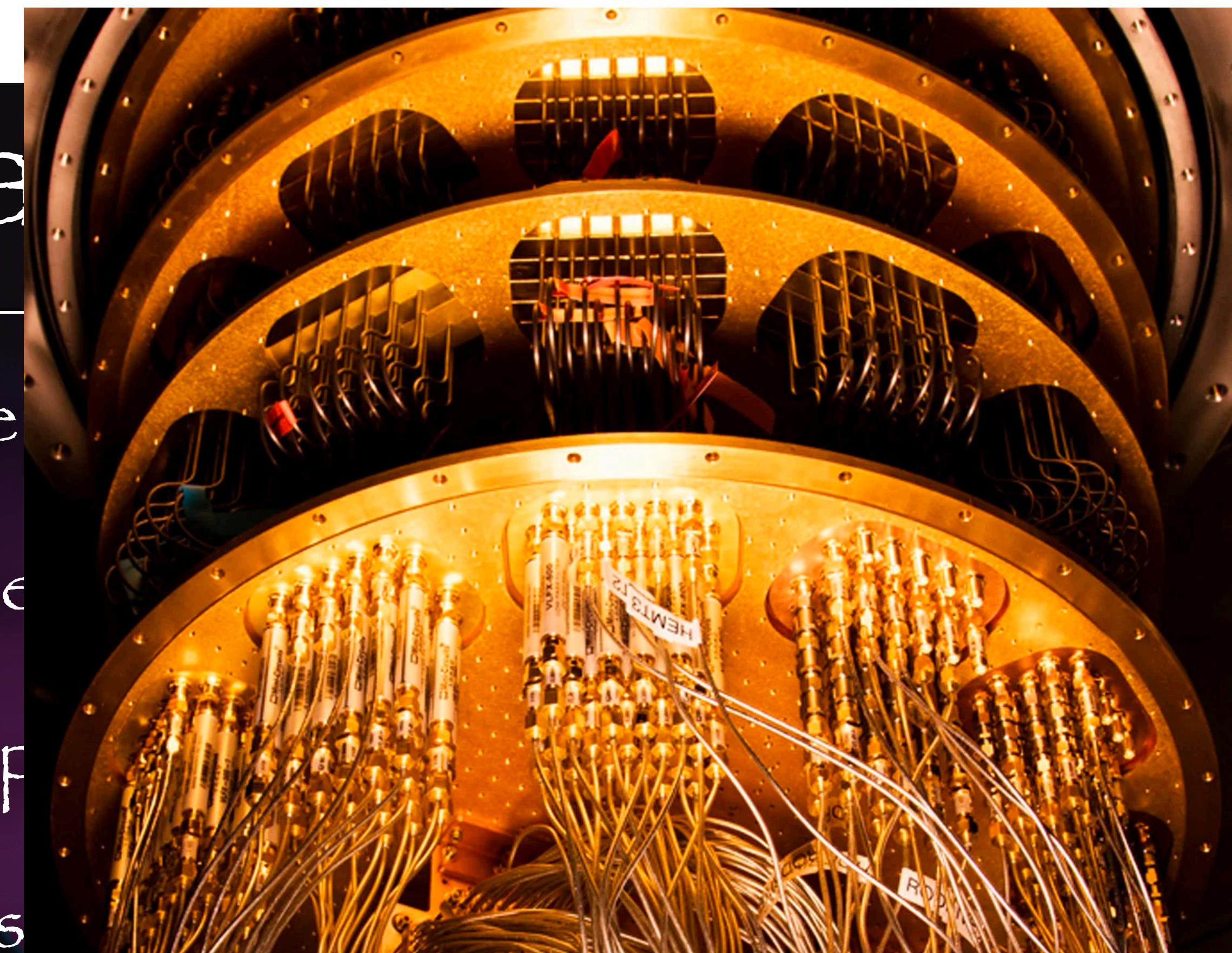
ChatGPT rebuttal for

Diffusion generates

NeR

Quantum Pre-trained Transformers (QPT) with perplexity 1.1??

ChatGPT summarizes



like?

&

Transformers?

aning?



rior:

e?



# What CVPR 2050 be like?

We haven't solved a **dog level** embodied AI yet!



AGI is just 5-10 years away!!

We haven't solved **compositionality** yet!

# 2050: An AI Odyssey

Prolog: what CVPR 2050 be like



Chapter 1: The Possible Impossibilities

Chapter 2: The Impossible Possibilities

Chapter 3: The Paradox

# The Possible Impossibilities?

AGI is seemingly around the corner;  
Is there really anything “impossible” with  
GPT5/6/7?

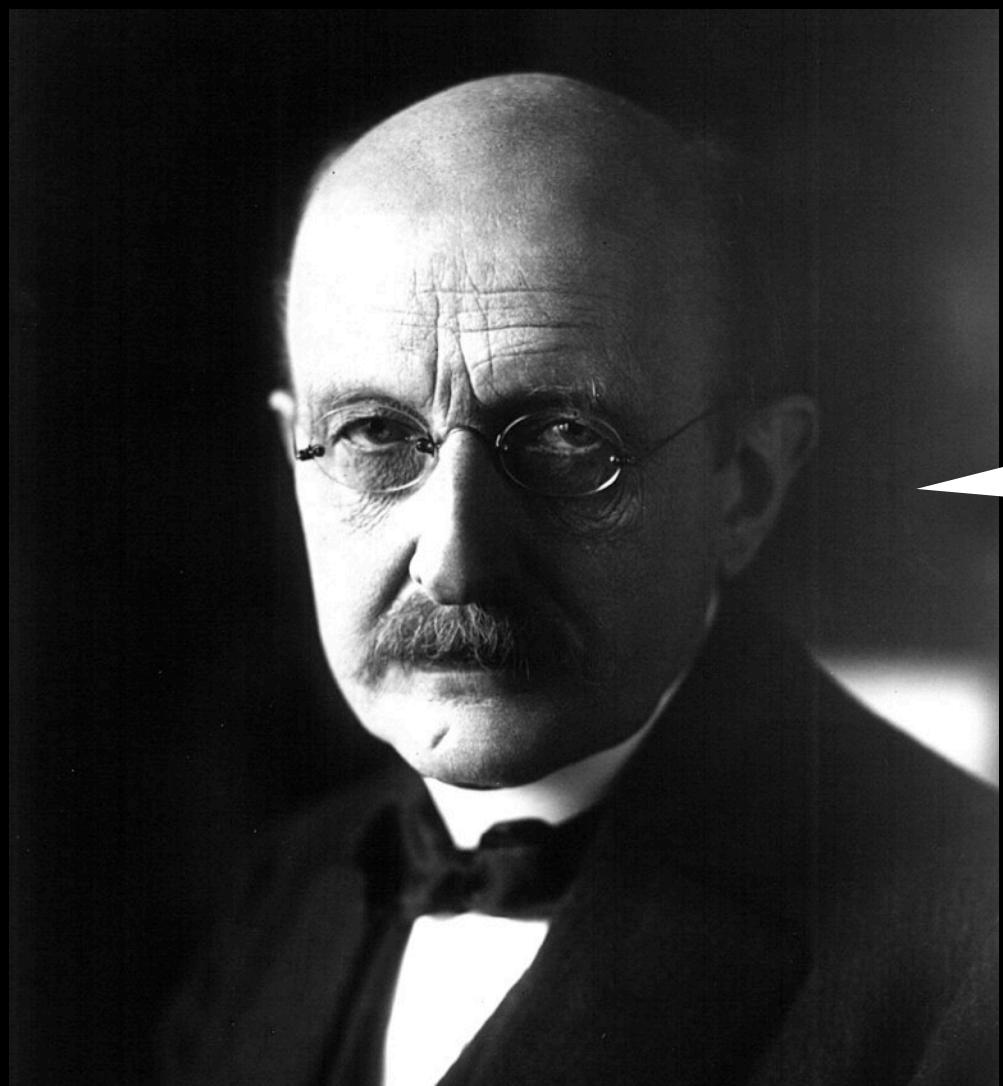
# Circa 1878 ...

Philipp von Jolly

"in this field, almost everything is already discovered, and all that remains is to fill a few unimportant holes"



Max Planck



"I don't wish to discover new things,  
only to u

fire 🔥 Quantum Physics 🔥

# Fast forward to the 20th/21st cent. ....

Dark Matter  
Schrödinger's cat  
Wave-particle duality  
Spacetime continuum

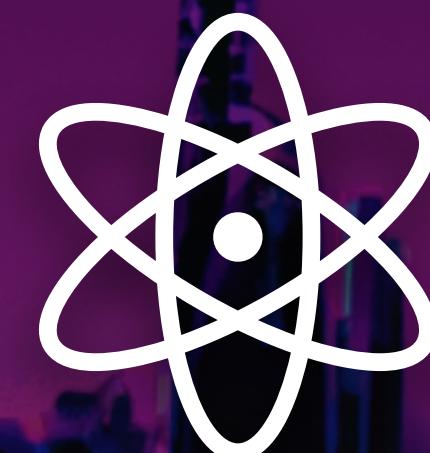
# Fast forward to the 20th/21st cent. ...

Dark Matter

Schrödinger's cat

Wave-particle duality

Spacetime continuum



Possible impossibilities  
Impossible possibilities  
Commonsense paradox  
Moravec's paradox  
Generative AI paradox

# The Possible Impossibilities?

In the limit,

- can AGI arrive without embodiment?
- can RLHF fully align LLMs to factuality?
- can Transformers truly master compositionally?

# Faith and Fate: Limits of Transformers on Compositionality

— arXiv:2305.18654 —

Nouha Dziri,



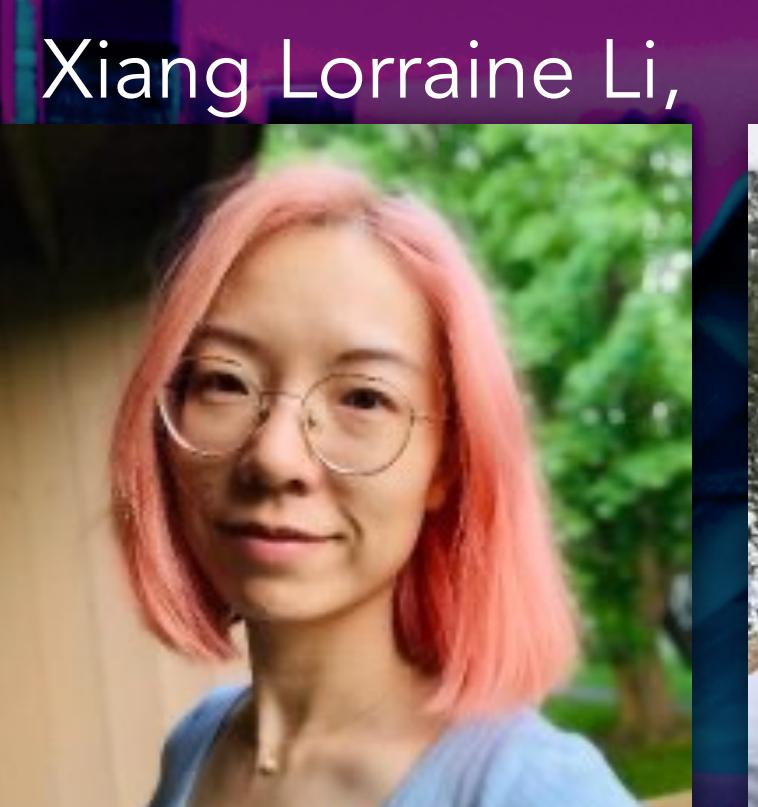
Ximing Lu,



Melanie Sclar



Xiang Lorraine Li,



Liwei Jiang,



Bill Yuchen Lin,



Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, Yejin Choi



# Does sparks of AGI (i.e., GPT4) solve multiplication?

GPT4



Multiply 999 by 867.



The product of multiplying 999 by 867 is 824,133.



The correct answer is  
866,133



Multiply 999 by 867. Think step by step



To multiply 999 by 867 step by step, we can use the traditional long multiplication method. Here's how it would look:

```
css
 999
 x 867
 -----
 8993  (999 times 7)
 8993  (999 times 6, shifted one place to the left)
 + 8993  (999 times 8, shifted two places to the left)
 -----
 866,733
```



The correct answer is  
866,133

Therefore, 999 multiplied by 867 equals 866,733.

# Does sparks of AGI (i.e., GPT4) solve multiplication?



GPT4

Multiply 999 by 867.



The product of multiplying 999 by 867 is 824,133.



The correct answer is  
866,133

**GPT4** achieves only **59% accuracy** on 3-digit × 3-digit multiplication.

**4% accuracy for 4x4 multiplication!**

**0% accuracy for 5x5 multiplication!**



**Kevin Patrick Murphy**  
@sirbayes

I like this paper. They prove that transformers are guaranteed to suffer from compounding errors when doing long reasoning chains (as @ylecun has argued), and much apparent "success" is just due to unreliable pattern matching / shortcut learning.



arxiv.org  
Faith and Fate: Limits of  
Transformer large language  
admiration for their exce

5:21 PM · Jun 15, 2023 · 266.1K Views

236 Retweets 12 Quotes 1,392 Likes 85



**yobibyte**  
@y0b1byte

Best paper I've read in a while. No excessive hype, clear question, rigorous empirical evaluation. Highly recommended.  
[arxiv.org/abs/2305.18654](https://arxiv.org/abs/2305.18654)



Cornell University

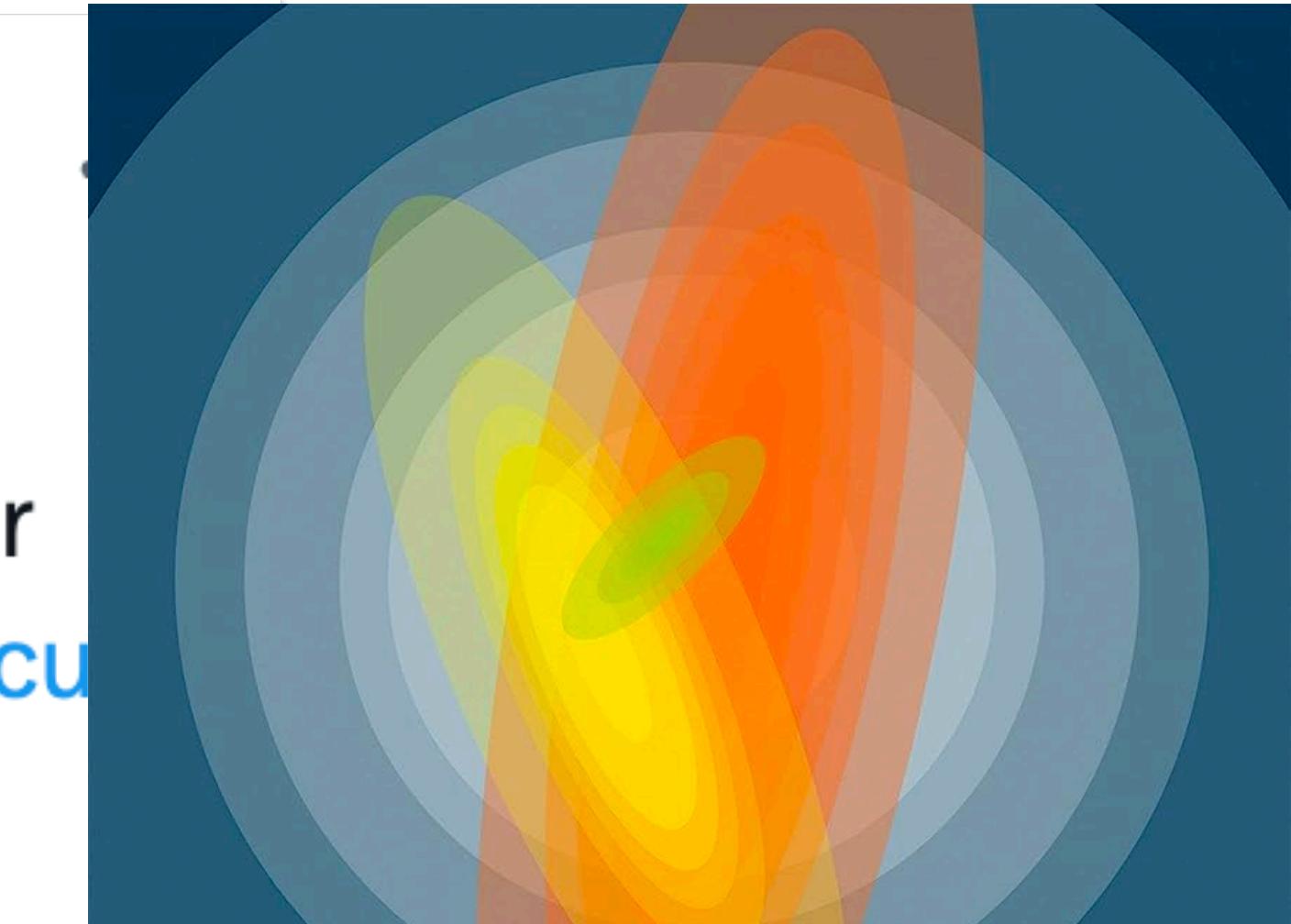
Computer Science > Computation and Language

arXiv:2305.18654 (cs)

[Submitted on 29 May 2023 (v1), last revised 1 Jun 2023 (this version, v2)]

## Faith and Fate: Limits of Transformers on Compositionality

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, Yejin Choi



Download PDF

# Does SUPERVISED TRAINING on 1.8 million examples\* solve multiplication? (W/ GPT3)

Davinci



Multiply 999 by 867.



The product is 865,233.



The correct answer is  
866,133

GPT3 fine-tuned achieves only **55% accuracy** on 3-digit x 3-digit multiplication.

*Exhaustively training on the task examples does not solve multiplication!*

Maybe tokenization is the issue ... ??

Let's train **GPT2** from scratch on multiplication data.

# SUPERVISED Training with Correct Tokenization will do it? (w/ GPT2-XL from scratch ...)

*Each digit is assigned to one token*

*Each math symbol (e.g. “=”, “x”) is assigned to one token*

*Train up to 4x4 (90M\*)*

GPT2-XL



Multiply 999 by 867.

The product is 865,233.



The correct answer is  
66,233

**Tokenization** does not seem the issue for the low performance!

**GPT2-XL** achieves only **10% accuracy** on 3-digit x 3-digit multiplication.

# It's just a matter of step-by-step? (w/ GPT4 + scratchpad\*)

$$\begin{array}{r} \times \quad 999 \\ 866 \\ \hline 5994 \\ 5994 \\ 7992 \\ \hline = \quad 865134 \end{array}$$

Accuracy increases to  
**92%** on 3x3  
multiplication!!



Multiply 999 by 866.

Let's perform the multiplication step by step:

Let's multiply 999 by the digit in the ones place of 866, which is 6.

1. Multiply 6 by the digit in the ones place of 999, which is 9.

This gives  $6 \times 9 = 54$ . Write down the result 4 and carry over the 5 to the next step.

2. ....

Let's multiply 999 by the digit in the tens place of 867, which is 6.

3. Multiply 7 by the digit in the ones place of 999, which is 9. This gives  $7 \times 9 = 63$ . Write down the result 3 and carry over the 6 to the next step.

4. ....



Multiply 999 by 867.

Let's perform the multiplication step by step:

Let's multiply 999 by the digit in the ones place of 867, which is 7. ....

.....

The final answer is **866,133**



\* We consider 5-shot 3x3 examples in each prompt.

# It's just a matter of step-by-step? (w/ GPT4 + scratchpad\*)

## 4x4 multiplication

Multiply 8999 by 1866.



Let's perform the multiplication step by step:

Let's multiply 8999 by the digit in the ones place of 1866, which is 6.

1. Multiply 6 by the digit in the ones place of 8999, which is 9.
2. ....

Accuracy decreases to 4%  
on 4x4 multiplication!!



GPT4

Let's perform the multiplication step by step:

.....

The final answer is 16,695,134.



## 5x5 multiplication

Multiply 78999 by 41866.



Let's perform the multiplication step by step:

Let's multiply 78999 by the digit in the ones place of 41866, which is 6.

1. Multiply 6 by the digit in the ones place of 78999, which is 9.
2. ....

Accuracy decreases to 2%  
on 5x5 multiplication!!



GPT4

Let's perform the multiplication step by step:

.....

The final answer is 3,305,251,134.



# Wait, didn't previous work said "transformers absolutely can learn true multi-step algorithms in-context"???



Thomas Miconi  
@ThomasMiconi

Interesting, but... Isn't that in opposition to Zhou et al. 2022?

Transformers absolutely can learn  
and generalize to arbitrary problem

	Method	Subtraction	Multiplication*
	Algorithmic prompt	65.6%	79.7%
	Best available baseline	16.7%	5.5%



Instead  
~~X~~

arxiv.org

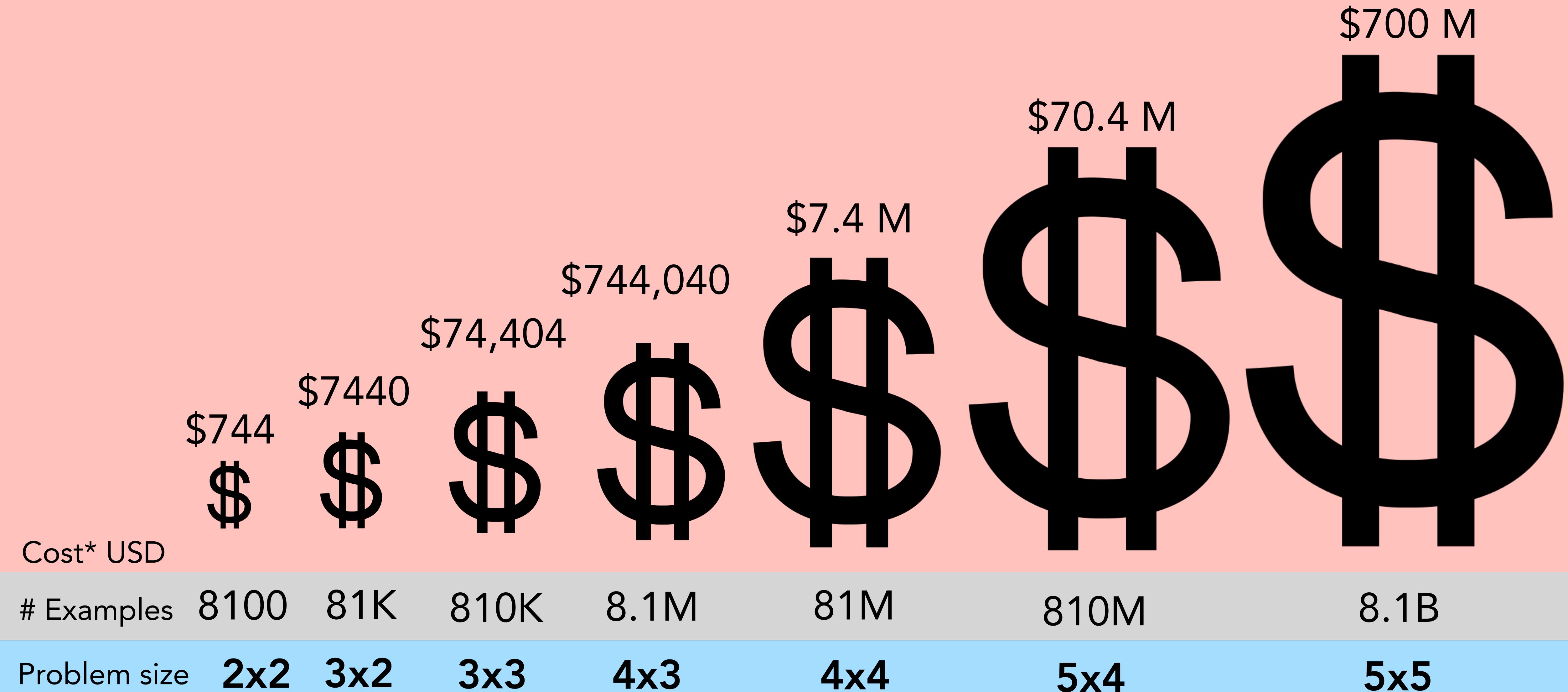
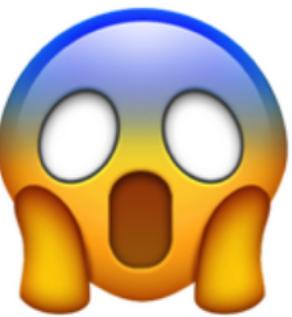
Teaching Algorithmic Reasoning via In-context Learning

We investigate the **fundamental limits** of achieving **full mastery** of the task rather than incremental improvements.

Partial Mastery

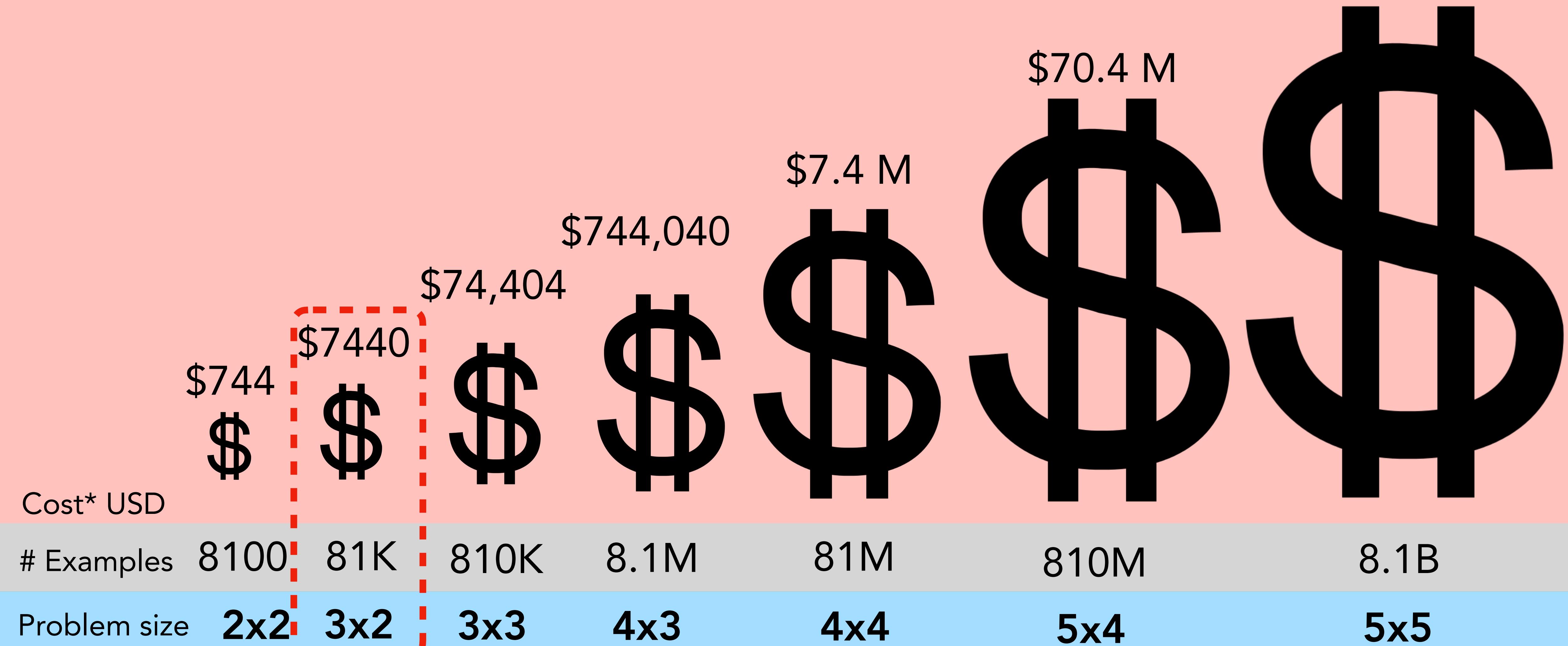
\*they report GPT4 doesn't do well when multiplying digits > 3, thus covert the problem manually to addition over small digit (<= 3) multiplications

# How about fine-tuning GPT3 on scratchpad?



\*Cost for 4 epochs with text-davinci-003

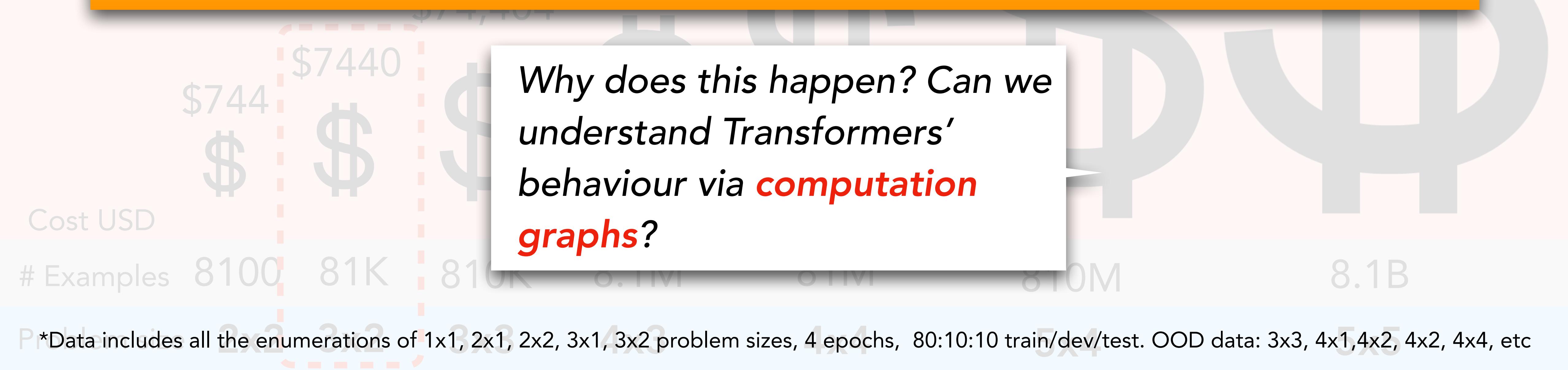
# How about fine-tuning GPT3 on scratchpad?



# How about fine-tuning\* GPT3 on scratchpad?

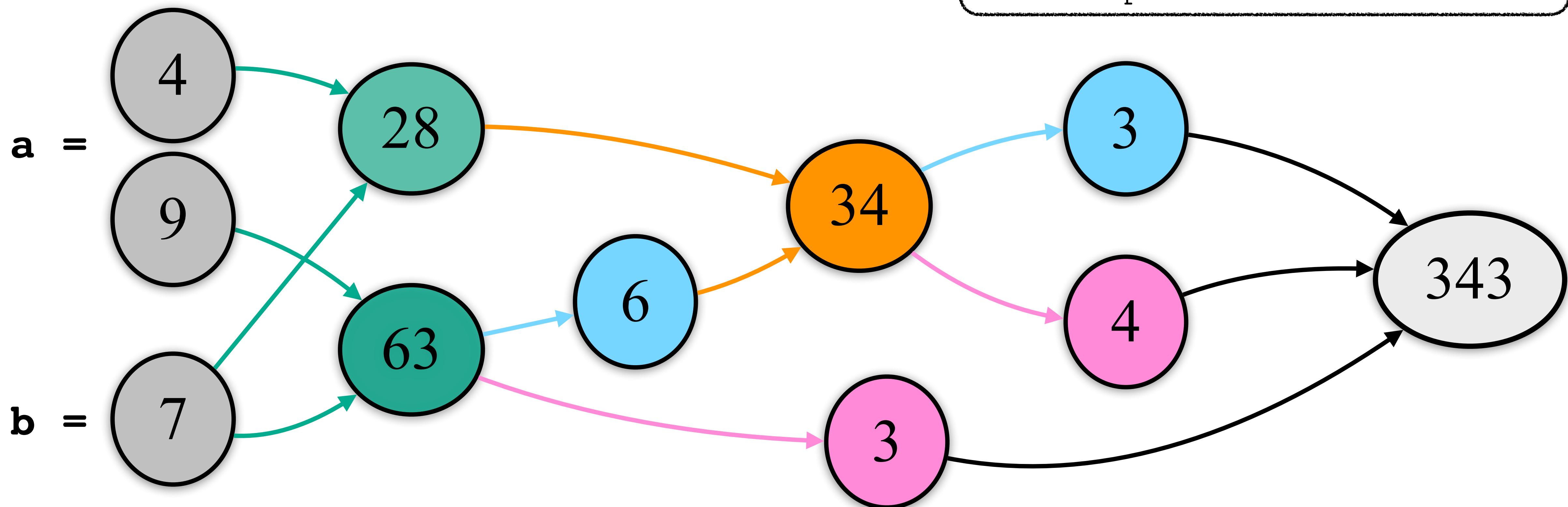
GPT3 achieves **96% accuracy** on in-distribution data but drops sharply to **zero** on OOD multiplication data.

*Why does this happen? Can we understand Transformers' behaviour via **computation graphs**?*

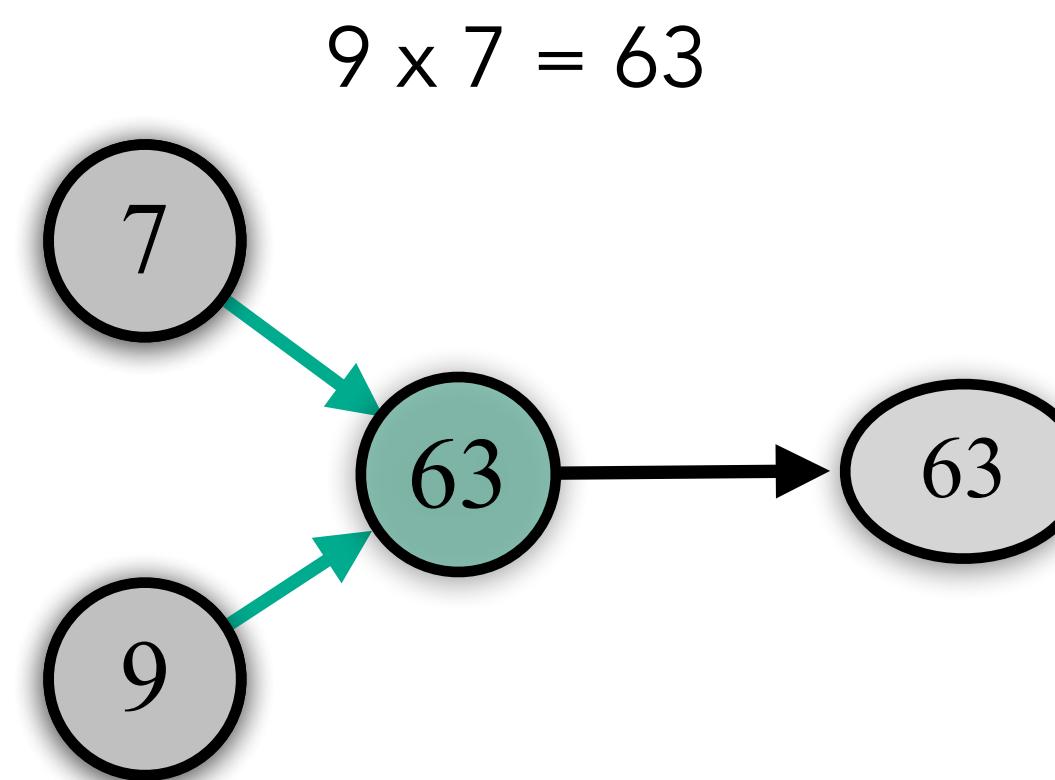


\*Data includes all the enumerations of 1x1, 2x1, 2x2, 3x1, 3x2 problem sizes, 4 epochs, 80:10:10 train/dev/test. OOD data: 3x3, 4x1, 4x2, 4x3, 4x4, etc

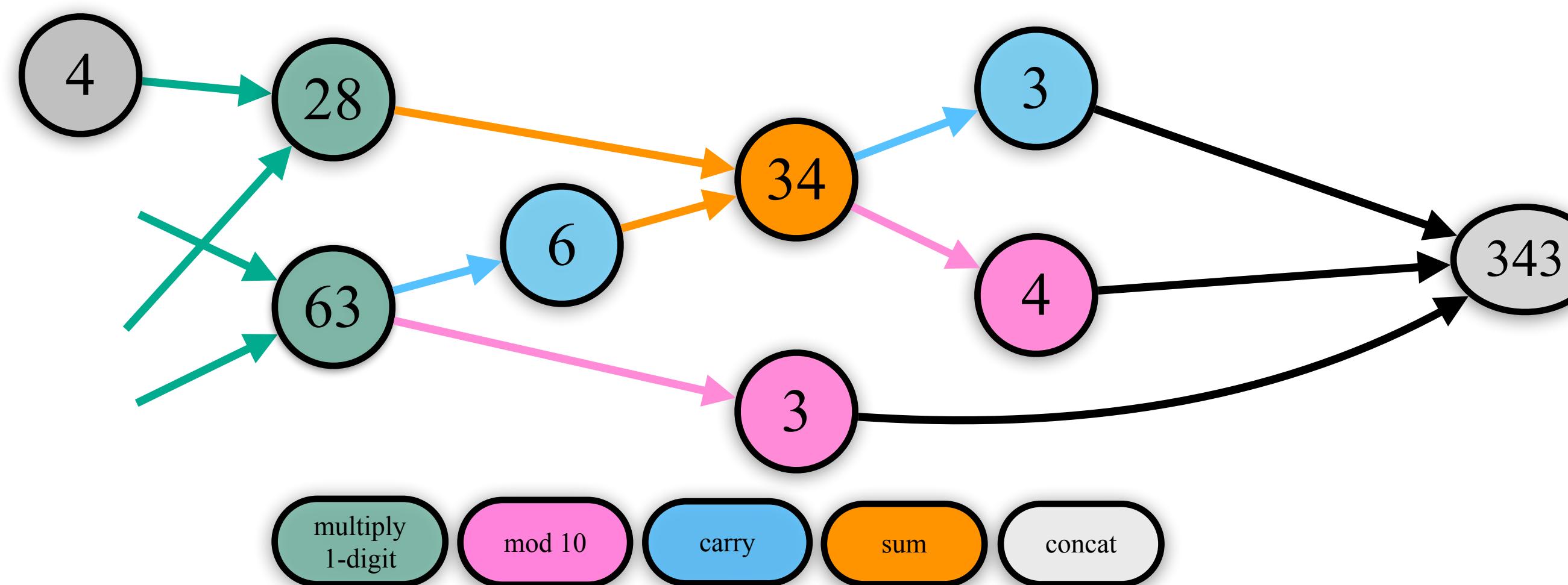
# Computation graph for $49 \times 7$



# Model Performance Decreases as Graph Complexity Increases



graph width = 1  
graph depth = 3  
avg. parallelism = 1.3



**Graph Complexity**

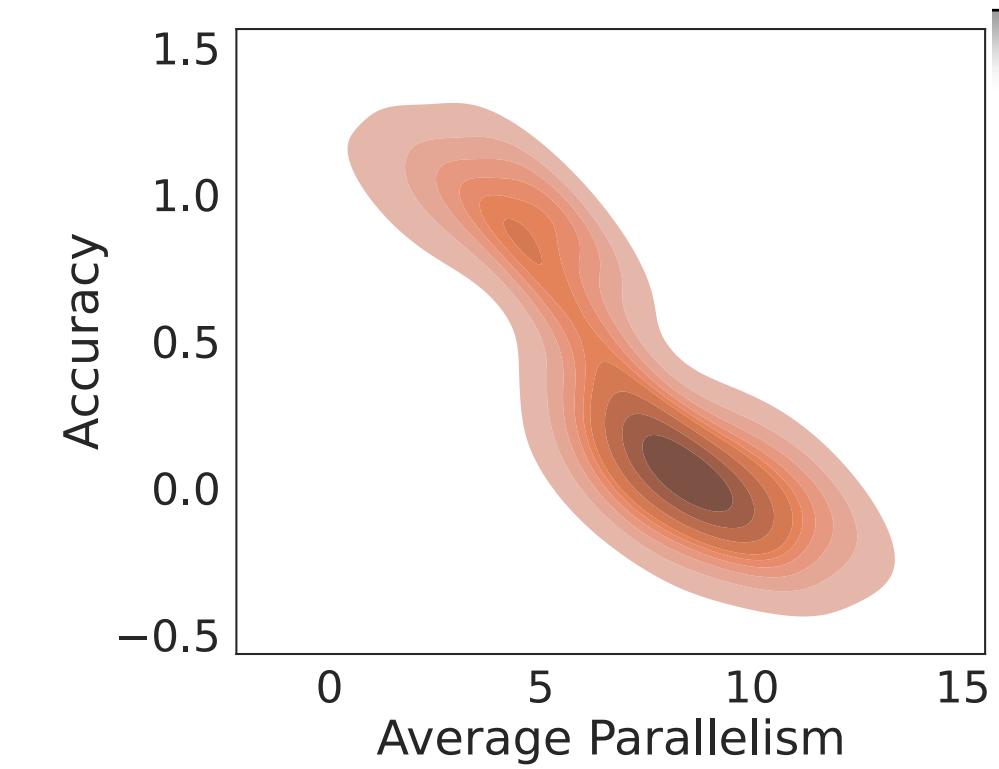
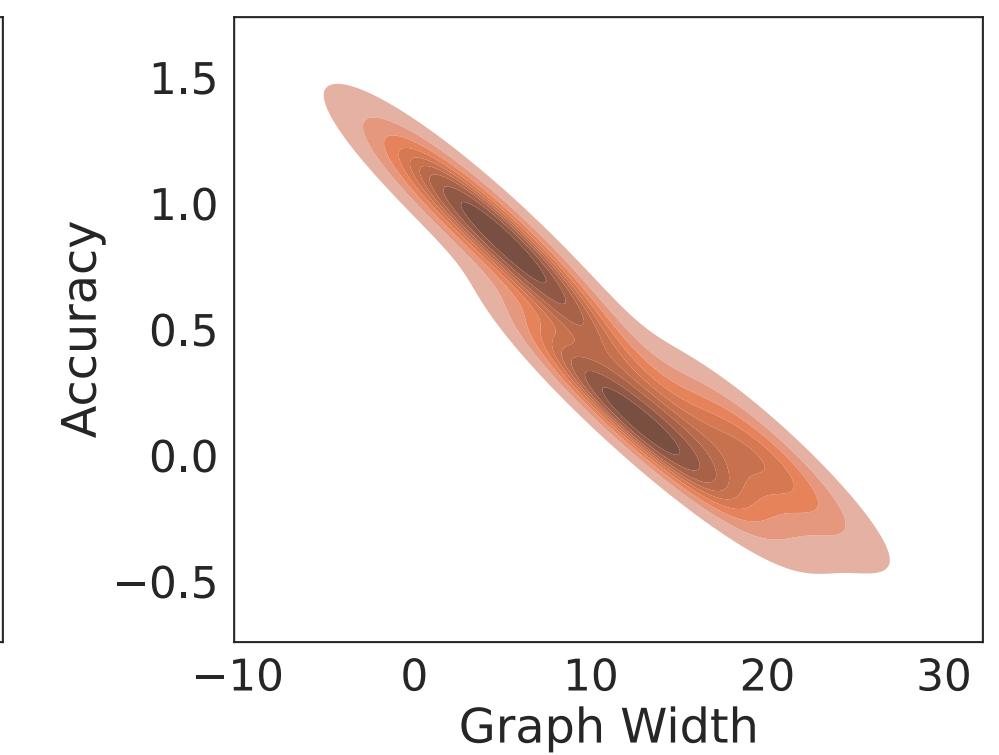
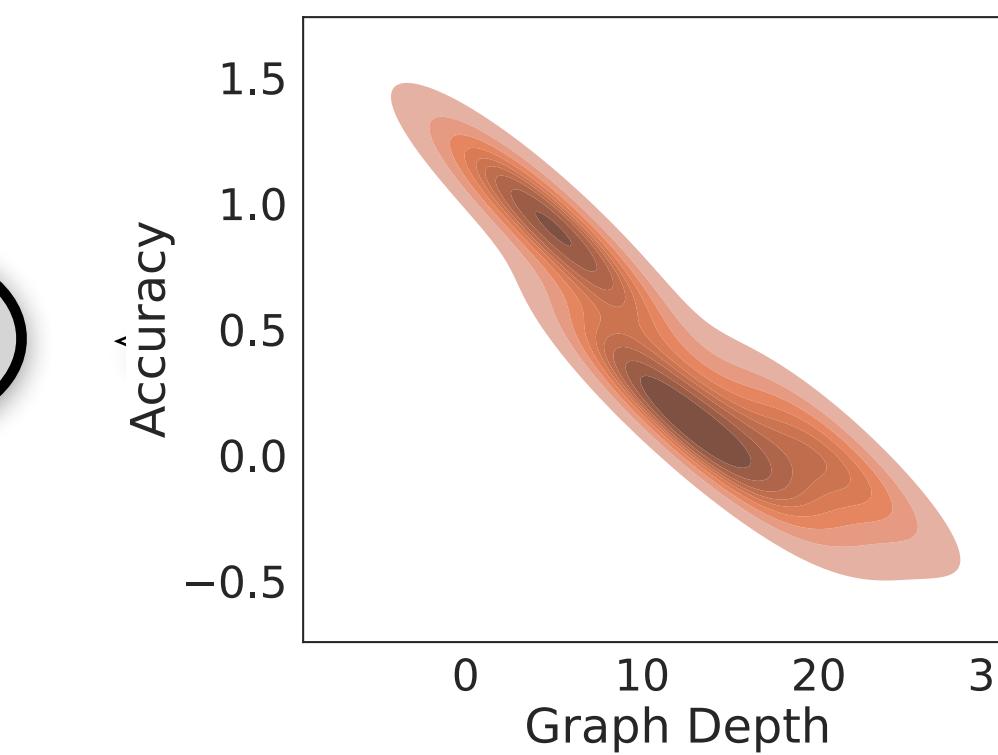
**graph width:** mode of  $\{d(v) : v \in V\}$

**graph depth:** the largest layer number in the graph

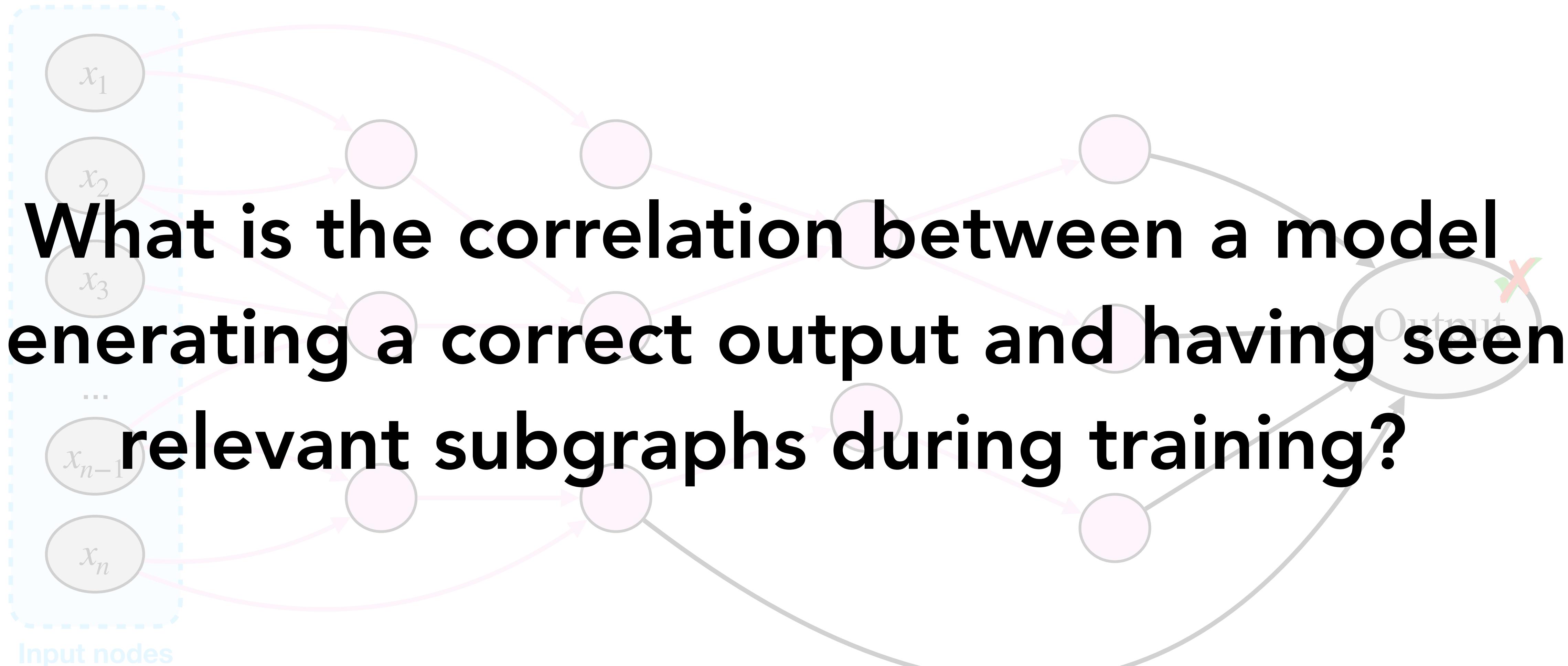
**average parallelism:** ratio between  $|V|$  and reasoning depth

GPT4 zero-shot

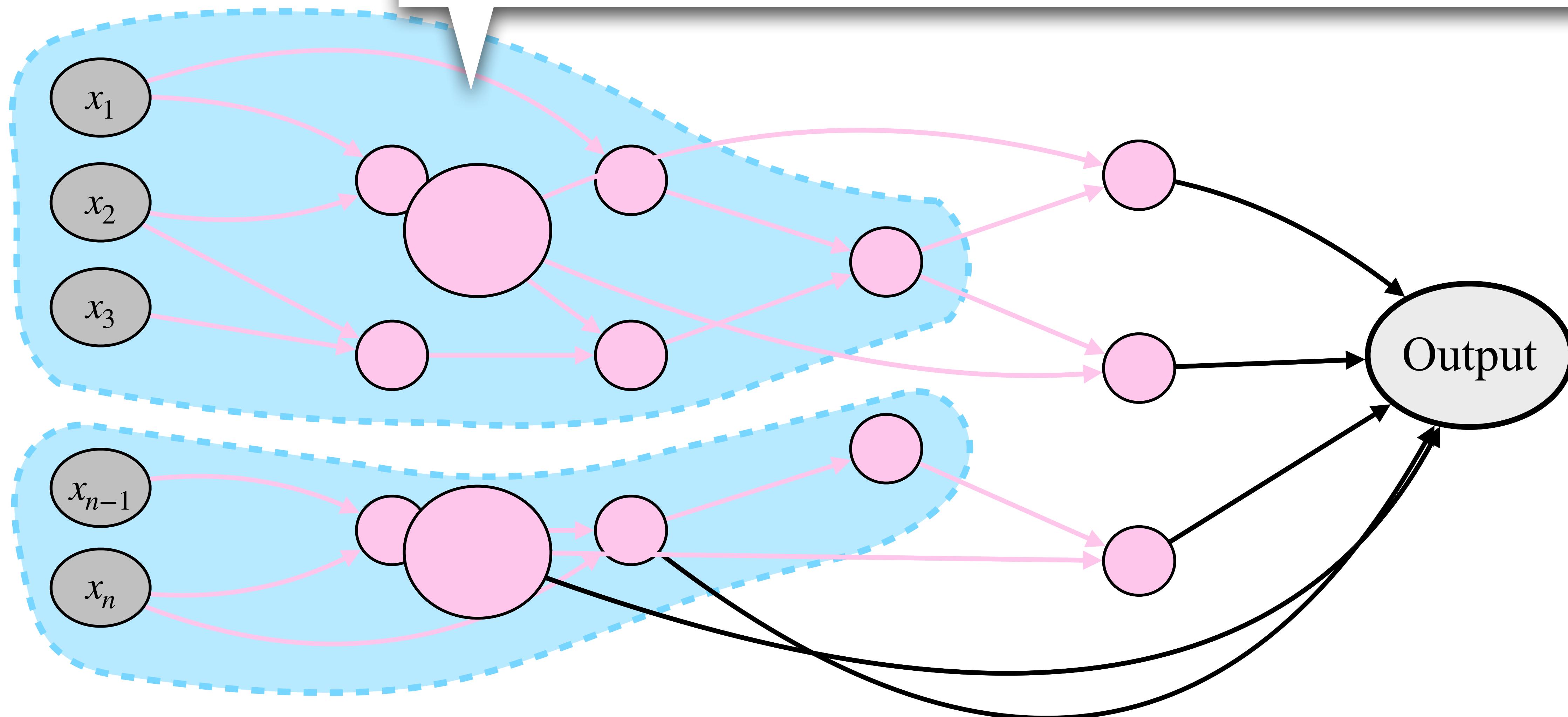
No. digits	1	2	3	4	5
1	1				
2		0.99			
3			0.97	0.59	
4	0.96	0.78	0.23	0.04	
5	0.91	0.54	0.09	0.01	0



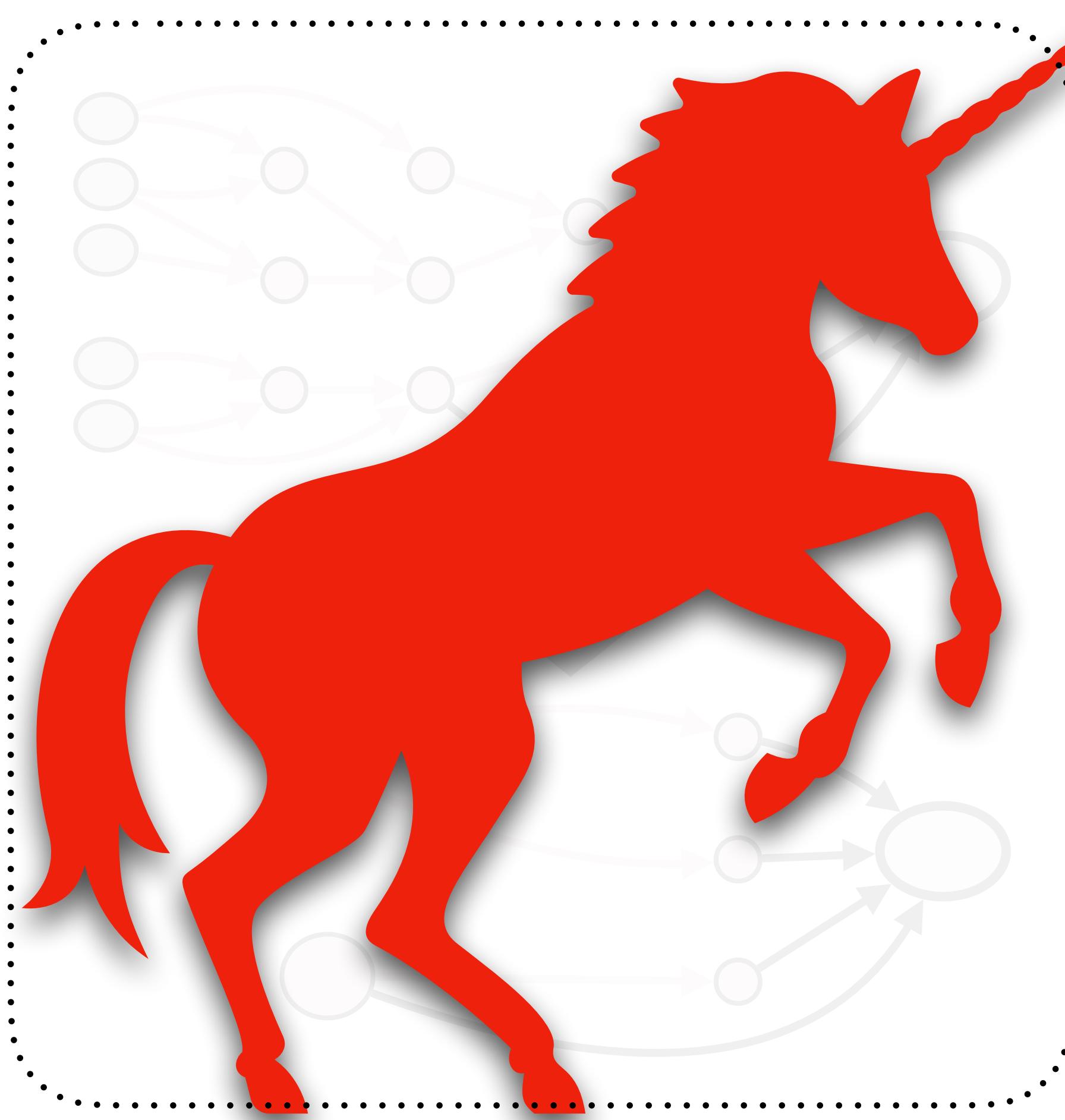
**What is the correlation between a model generating a correct output and having seen relevant subgraphs during training?**



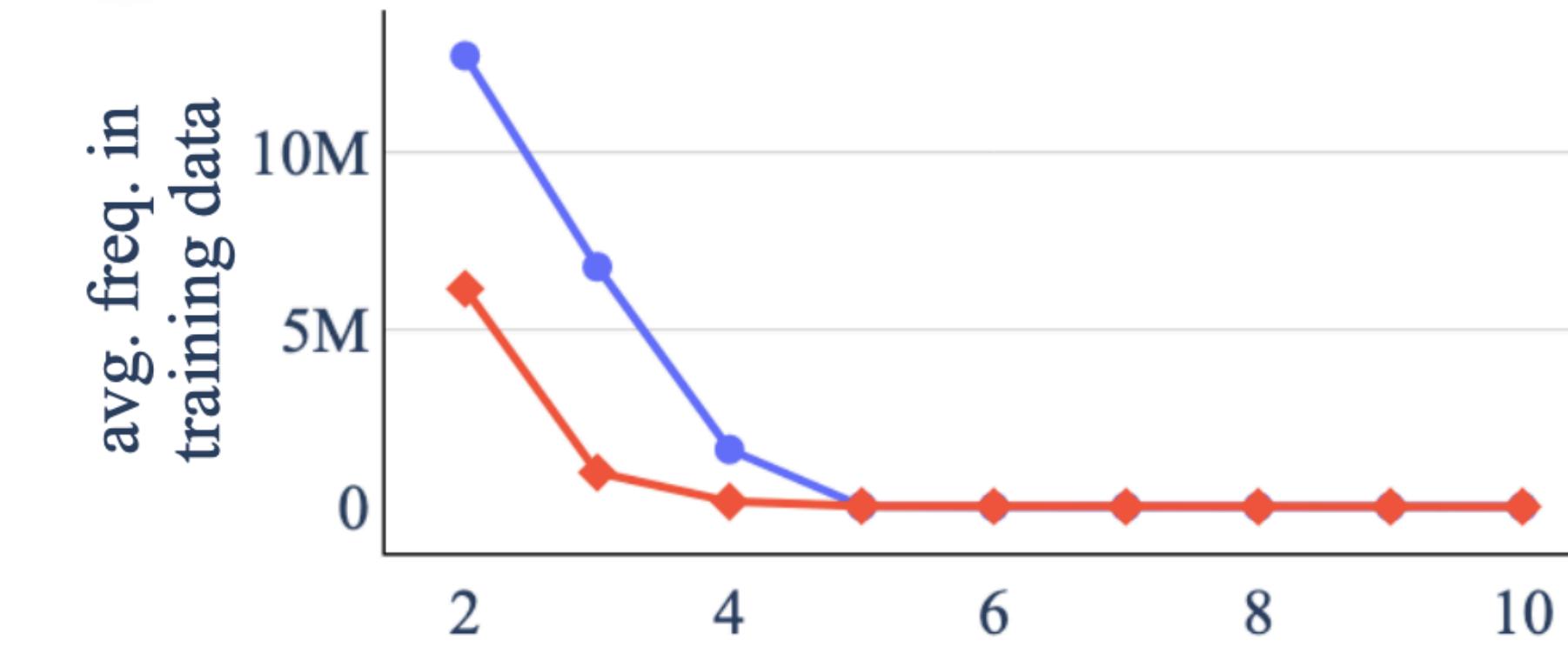
Detect subgraphs already seen during training: Want subgraphs during training, the inference is only **seemingly** highly compositional



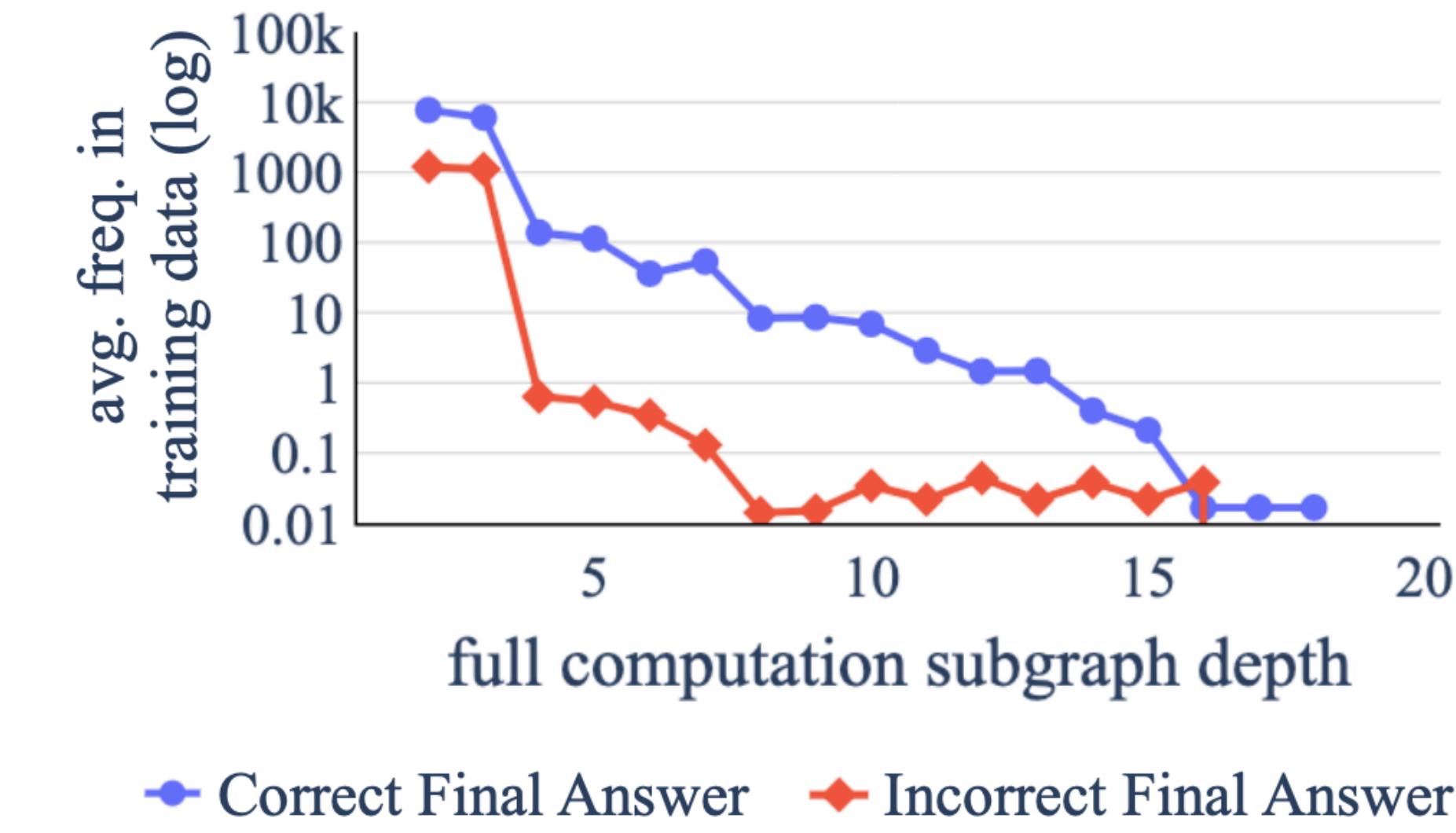
*Transformers' successes are heavily linked to having seen significant portions of the required computation graph during training*



Fine-tuned GPT3 - Multiplication

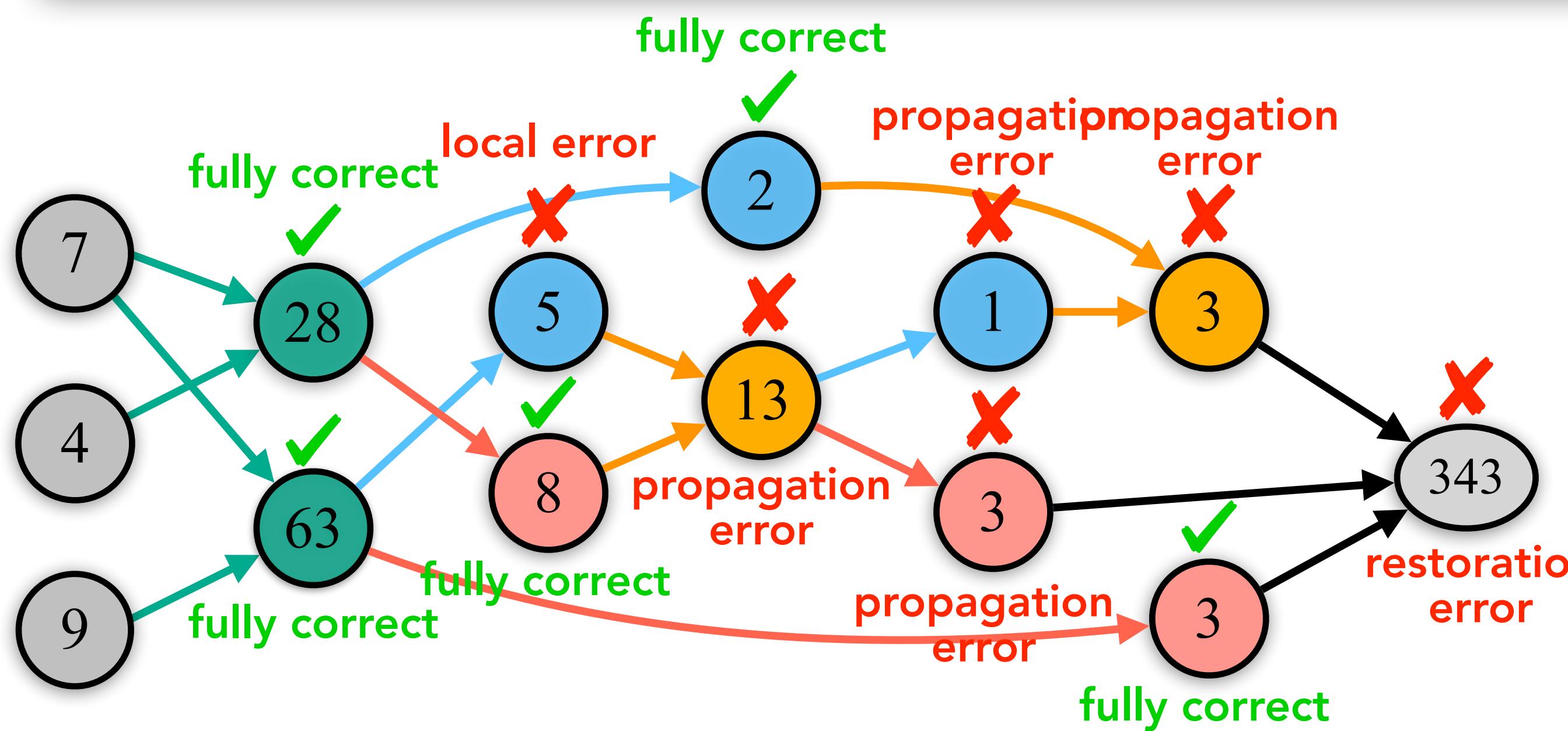


Fine-tuned GPT3 - Dynamic Programming



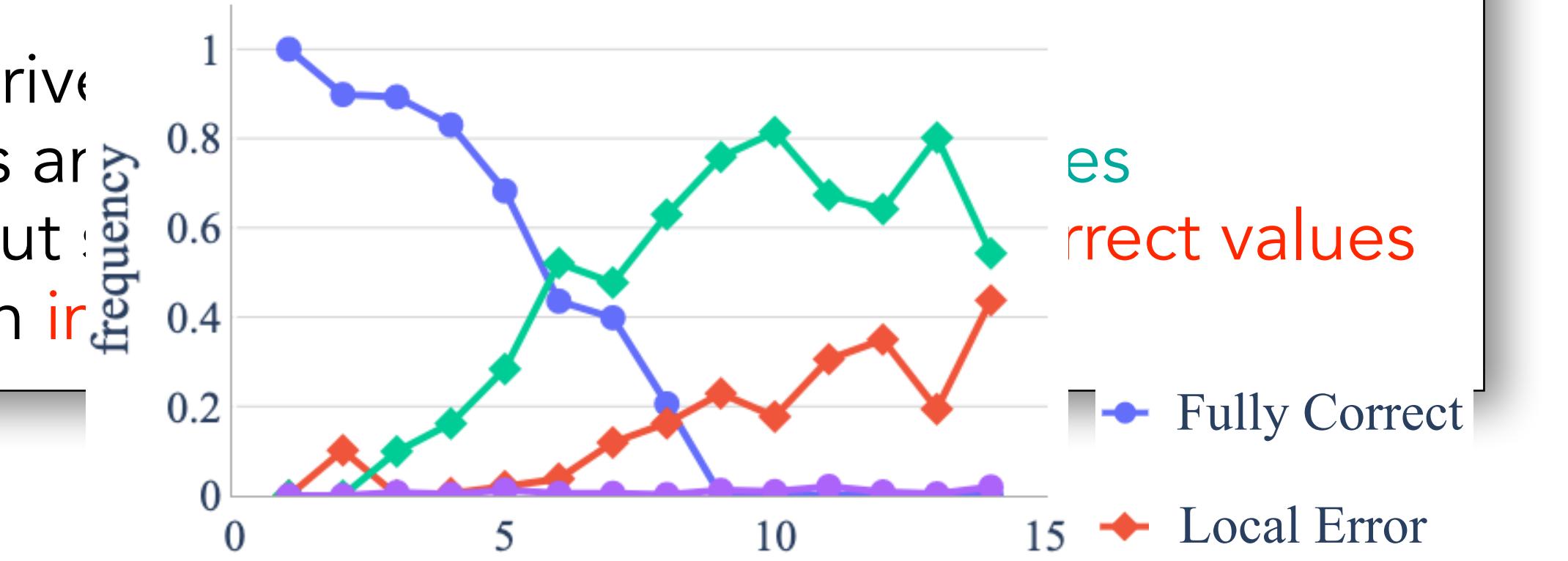
# What Types of Errors do Transformers Make at Different Reasoning Depths?

- Fully Correct:** v and ancestors have **correct values** and are derived from **correct computations**.
- Local Error:** v is derived from an **incorrect computation** but its ancestors have **correct values**.
- Propagation Error:** v is derived from a **correct computation** but its ancestors have **incorrect values**.
- Restoration Error:** v has a **correct value** but is derived from an **incorrect computation**.

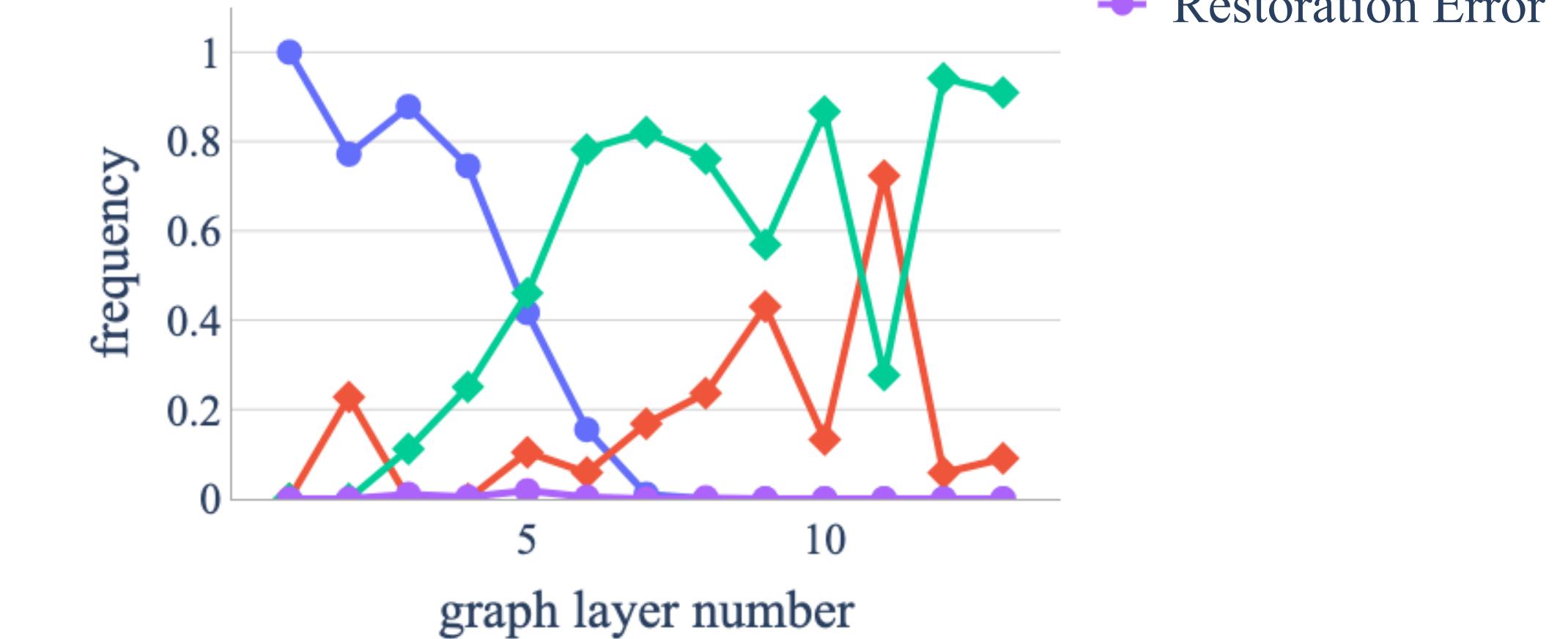


## Error Type

Five-shot GPT4 – Multiplication



Fine-tuned GPT3 – Multiplication



# Transformers' performance will rapidly decay with increased task complexity

## D Theoretical Results: Derivations

### D.1 Transformers struggle with problems with increasingly larger parallelism (width)

**Proposition D.1.** Let  $f_n(\mathbf{x}) = h_n(g(\mathbf{x}, 1), g(\mathbf{x}, 2), \dots, g(\mathbf{x}, n))$ . Let  $\hat{h}_n, \hat{g}, \hat{f}_n$  be estimators of  $h_n, g, f_n$  respectively. Assume  $\mathbb{P}(h_n = \hat{h}_n) = 1$  and  $\mathbb{P}(h_n(X) = h_n(Y) \mid X \neq Y) < \beta\alpha^n$  for some  $\alpha \in (0, 1)$  and  $\beta > 0$  (i.e.  $\hat{h}_n$  perfectly estimates  $h_n$ , and  $h_n$  is almost injective). If  $\mathbb{P}(g \neq \hat{g}) = \epsilon > 0$  and errors in  $\hat{g}$  are independent, then  $\lim_{n \rightarrow +\infty} \mathbb{P}(f_n \neq \hat{f}_n) = 1$ .

*Proof.* For ease of writing, let  $X_i = g(X, i)$  and  $Y_i = \hat{g}(X, i)$ , and let  $\mathbf{X} = (X_1, \dots, X_n)$  and  $\mathbf{Y} = (Y_1, \dots, Y_n)$ . We will compute some auxiliary probabilities, and then upper bound  $\mathbb{P}(f = \hat{f})$ , to finally compute its limit.

$$\begin{aligned} \mathbb{P}(\mathbf{X} = \mathbf{Y}) &= \mathbb{P}(X_1 = Y_1, X_2 = Y_2, \dots, X_n = Y_n) \\ &= \mathbb{P}(X_1 = Y_1) \cdot \mathbb{P}(X_2 = Y_2) \cdots \mathbb{P}(X_n = Y_n) = \mathbb{P}(g = \hat{g})^n = (1 - \epsilon)^n \end{aligned} \quad (2)$$

Since by hypothesis we know  $\mathbb{P}(h_n(\mathbf{Y}) = \hat{h}_n(\mathbf{Y})) = 1$ , we have that:

$$\begin{aligned} \mathbb{P}(h_n(\mathbf{X}) = \hat{h}_n(\mathbf{Y}) \mid \mathbf{X} \neq \mathbf{Y}) &= \mathbb{P}(h_n(\mathbf{X}) = \hat{h}_n(\mathbf{Y}) \cap h_n(\mathbf{Y}) = \hat{h}_n(\mathbf{Y}) \mid \mathbf{X} \neq \mathbf{Y}) \\ &= \mathbb{P}(h_n(\mathbf{X}) = h_n(\mathbf{Y}) = \hat{h}_n(\mathbf{Y}) \mid \mathbf{X} \neq \mathbf{Y}) \\ &\leq \mathbb{P}(h_n(\mathbf{X}) = h_n(\mathbf{Y}) \mid \mathbf{X} \neq \mathbf{Y}) \\ &< \beta\alpha^n \end{aligned} \quad (3)$$

We will now estimate  $\mathbb{P}(f_n = \hat{f}_n)$  using the law of total probability w.r.t. the event  $\mathbf{X} = \mathbf{Y}$ .

$$\begin{aligned} \mathbb{P}(f_n = \hat{f}_n) &= \mathbb{P}(h_n(\mathbf{X}) = \hat{h}_n(\mathbf{Y})) \\ &= \mathbb{P}(h_n(\mathbf{X}) = \hat{h}_n(\mathbf{Y}) \mid \mathbf{X} = \mathbf{Y}) \cdot \mathbb{P}(\mathbf{X} = \mathbf{Y}) + \mathbb{P}(h_n(\mathbf{X}) = \hat{h}_n(\mathbf{Y}) \mid \mathbf{X} \neq \mathbf{Y}) \cdot \mathbb{P}(\mathbf{X} \neq \mathbf{Y}) \\ &= \mathbb{P}(h_n(\mathbf{X}) = \hat{h}_n(\mathbf{X})) \cdot \mathbb{P}(\mathbf{X} = \mathbf{Y}) + \mathbb{P}(h_n(\mathbf{X}) = \hat{h}_n(\mathbf{Y}) \mid \mathbf{X} \neq \mathbf{Y}) \cdot (1 - \mathbb{P}(\mathbf{X} = \mathbf{Y})) \\ &= 1 \cdot (1 - \epsilon)^n + \mathbb{P}(h_n(\mathbf{X}) = \hat{h}_n(\mathbf{Y}) \mid \mathbf{X} \neq \mathbf{Y}) \cdot (1 - (1 - \epsilon)^n) \quad (\text{using 2 and hypothesis}) \\ &< (1 - \epsilon)^n + \beta\alpha^n \cdot (1 - (1 - \epsilon)^n) \quad (\text{using 3}) \\ &< \beta\alpha^n + (1 - \epsilon)^n \cdot (1 - \beta\alpha^n) \end{aligned}$$

To conclude our proof, we will show that  $\lim_{n \rightarrow +\infty} \mathbb{P}(f_n = \hat{f}_n)$  exists and compute its value. Note that since  $1 - \epsilon \in [0, 1]$  and  $\alpha \in (0, 1)$ , trivially  $\lim_{n \rightarrow +\infty} \beta\alpha^n + (1 - \epsilon)^n \cdot (1 - \beta\alpha^n) = 0$ .

$$0 \leq \liminf_{n \rightarrow +\infty} \mathbb{P}(f_n = \hat{f}_n) \leq \limsup_{n \rightarrow +\infty} \mathbb{P}(f_n = \hat{f}_n) \leq \limsup_{n \rightarrow +\infty} \beta\alpha^n + (1 - \epsilon)^n \cdot (1 - \beta\alpha^n) = 0$$

Then,  $\lim_{n \rightarrow +\infty} \mathbb{P}(f_n = \hat{f}_n) = 0$  and we conclude  $\lim_{n \rightarrow +\infty} \mathbb{P}(f_n \neq \hat{f}_n) = 0$ .  $\square$

**Corollary D.1.** Assume that a model  $\mathcal{M}$  solves shifted addition perfectly, but it incorrectly solves at least one  $m$  digit by 1 digit multiplication for some fixed  $m$ . Then, the probability that  $\mathcal{M}$  will solve any  $m$  digit by  $n$  digit multiplication using the long-form multiplication algorithm tends to 0.

*Proof.* We define  $s : \mathbb{Z}_{10}^{m+n} \times \mathbb{N} \rightarrow \mathbb{N} \times \mathbb{N}$ ,  $d : \mathbb{N} \times \mathbb{Z}_{10} \rightarrow \mathbb{N}$ ,  $h_n : \mathbb{N}^n \rightarrow \mathbb{N}$ , and  $f_n : \mathbb{Z}_{10}^{m+n} \rightarrow \mathbb{N}$  as follows.

$$s([x_1, \dots, x_m, x_{m+1}, \dots, x_{m+n}], j) := (x_1 \widehat{\cdot} x_2 \widehat{\cdot} \dots \widehat{\cdot} x_m, x_{m+j}) \quad \text{where } x_1 \widehat{\cdot} x_2 \widehat{\cdot} \dots \widehat{\cdot} x_m \text{ denotes concatenating digits } x_i$$

$$\begin{aligned} d(x, y) &:= x \cdot y \\ g &:= d \circ s \end{aligned}$$

$$h_n(x_1, \dots, x_n) := \sum_{i=1}^n x_i 10^{n-i}$$

$$f_n(\mathbf{x}) := h_n(g(\mathbf{x}, 1), g(\mathbf{x}, 2), \dots, g(\mathbf{x}, n))$$

Note that  $g$  defines the base-10 multiplication between  $m$ -digit numbers  $(x_1 x_2 \dots x_m)$  and 1-digit numbers  $(x_{m+j})$ , where  $s$  denotes the selection of the numbers to multiply and  $d$  denotes the actual multiplication. Note that  $h_n$  describes the shifted addition used at the end of long-form multiplication to combine  $n$   $m$ -digit by 1-digit multiplications. Therefore,  $f_n$  describes the long-form multiplication of  $m$ -digit by  $n$ -digit numbers.

By hypothesis,  $\mathbb{P}(g \neq \hat{g}) = \epsilon > 0$  and  $\mathbb{P}(h_n = \hat{h}_n) = 1$ , where  $\hat{g}$  and  $\hat{h}_n$  denote estimators using model  $\mathcal{M}$ . It can be shown that  $\mathbb{P}(h_n(X) = h_n(Y) \mid X \neq Y) < \beta\alpha^n$  for  $\alpha = 0.1$  and  $\beta = 10^m$ . Using Lemma D.1,  $\lim_{n \rightarrow +\infty} \mathbb{P}(f_n \neq \hat{f}_n) = 1$ , which concludes our proof.  $\square$

Note that Lemma D.1's proofs gives us empirical bounds once  $\epsilon$  and  $\alpha$  are approximated. Also note that our definition of  $g$  in the proof of Corollary D.1 highlights two possible sources of exponentially-accumulating error: errors in the selection of the numbers to multiply  $s$ , and errors in the actual  $m$ -digit by 1-digit multiplication  $d$ .

### D.2 Transformers struggle with problems that require increasingly larger iterative applications of a function (depth)

**Proposition D.2.** Let  $f_n(\mathbf{x}) = g^n(\mathbf{x})$ . Assume  $\mathbb{P}(g(X) = \hat{g}(Y) \mid X \neq Y) \leq c$  (i.e. recovering from a mistake due to the randomness of applying the estimator on an incorrect input has probability at most  $c$ ). If  $\mathbb{P}(g \neq \hat{g}) = \epsilon > 0$  with  $c + \epsilon < 1$ , then  $\liminf_{n \rightarrow +\infty} \mathbb{P}(f_n \neq \hat{f}_n) = 1 - \frac{c}{c + \epsilon}$ .

*Proof.* We first derive a recursive upper bound using the law of total probability, and then prove a non-recursive upper bound by induction.

$$\begin{aligned} s_n &:= \mathbb{P}(f_n = \hat{f}_n) = \mathbb{P}(g(g^{n-1}(Z)) = \hat{g}(g^{n-1}(Z))) \\ &= \mathbb{P}(g(\mathbf{X}) = \hat{g}(\mathbf{Y})) \quad \text{where } \mathbf{X} := g^{n-1}(Z) \text{ and } \mathbf{Y} := \hat{g}^{n-1}(Z) \\ &= \mathbb{P}(g(\mathbf{X}) = \hat{g}(\mathbf{Y}) \mid \mathbf{X} = \mathbf{Y}) \cdot \mathbb{P}(\mathbf{X} = \mathbf{Y}) + \mathbb{P}(g(\mathbf{X}) = \hat{g}(\mathbf{Y}) \mid \mathbf{X} \neq \mathbf{Y}) \cdot \mathbb{P}(\mathbf{X} \neq \mathbf{Y}) \\ &= \mathbb{P}(g(\mathbf{X}) = \hat{g}(\mathbf{X})) \cdot \mathbb{P}(\mathbf{X} = \mathbf{Y}) + \mathbb{P}(g(\mathbf{X}) = \hat{g}(\mathbf{Y}) \mid \mathbf{X} \neq \mathbf{Y}) \cdot (1 - \mathbb{P}(\mathbf{X} = \mathbf{Y})) \\ &= \mathbb{P}(g(\mathbf{X}) = \hat{g}(\mathbf{X})) \cdot s_{n-1} + \mathbb{P}(g(\mathbf{X}) = \hat{g}(\mathbf{Y}) \mid \mathbf{X} \neq \mathbf{Y}) \cdot (1 - s_{n-1}) \\ &\leq (1 - \epsilon) \cdot s_{n-1} + c \cdot (1 - s_{n-1}) \\ &\leq (1 - \epsilon - c) \cdot s_{n-1} + c \end{aligned}$$

We know  $s_1 = (1 - \epsilon)$  since  $s_1 = \mathbb{P}(f_1 = \hat{f}_1) = \mathbb{P}(g = \hat{g})$ . Let  $b := 1 - \epsilon - c$  for ease of writing. Then, we have

$$s_n \leq b \cdot s_{n-1} + c \quad (4)$$

It can be easily shown by induction that  $s_n \leq b^{n-1}(1 - \epsilon) + c \sum_{i=0}^{n-2} b^i$ :

- The **base case**  $n = 2$  is true since we know  $s_2 \leq b \cdot s_1 + c$ , and  $b \cdot s_1 + c = b(1 - \epsilon) + c = b^{2-1}(1 - \epsilon) + c \sum_{i=0}^{2-2} b^i$ , thus showing  $s_2 \leq b^{2-1}(1 - \epsilon) + c \sum_{i=0}^{2-2} b^i$
- The **inductive step** yields directly using Equation 4,

$$\begin{aligned} s_n &\leq b \cdot s_{n-1} + c \\ &\leq b \cdot \left( b^{n-2}(1 - \epsilon) + c \sum_{i=0}^{n-3} b^i \right) + c \leq b^{n-1}(1 - \epsilon) + c \sum_{i=1}^{n-2} b^i + c \leq b^{n-1}(1 - \epsilon) + c \sum_{i=0}^{n-2} b^i \end{aligned}$$

We can rewrite the geometric series  $\sum_{i=0}^{n-2} b^i$  in its closed form  $\frac{1-b^{n-1}}{1-b}$ , and recalling  $b := 1 - \epsilon - c$ ,

$$\begin{aligned} s_n &\leq b^{n-1}(1 - \epsilon) + c \frac{1 - b^{n-1}}{1 - b} = b^{n-1}(1 - \epsilon) + c \frac{1 - b^{n-1}}{c + \epsilon} \\ &= b^{n-1}(1 - \epsilon) + \frac{c}{c + \epsilon} - b^{n-1} \frac{c}{c + \epsilon} \\ &= b^{n-1} \left( 1 - \epsilon - \frac{c}{c + \epsilon} \right) + \frac{c}{c + \epsilon} \end{aligned}$$

# Shortcut Learning in Deep Neural Networks

Robert Geirhos<sup>1,2,\*,\$</sup>, Jörn-Henrik Jacobsen<sup>3,\*</sup>, Claudio Michaelis<sup>1,2,\*</sup>,  
Richard Zemel<sup>†,3</sup>, Wieland Brendel<sup>†,1</sup>, Matthias Bethge<sup>†,1</sup> & Felix A. Wichmann<sup>†,1</sup>

## Transformers Learn Shortcuts to Automata

By and large, the prior work was based on weaker LLMs, thus some might have wondered with extreme-scale, these problems magically go away

Ruixiang Tang<sup>†</sup>, Dehan Kong<sup>‡</sup>, Longtao Huang<sup>‡</sup>, Hui Xue<sup>‡</sup>

## Shortcut Learning of Large Language Models in Natural Language Understanding

Mengnan Du  
New Jersey Institute of Technology  
Newark, NJ, USA  
mengnan.du@njit.edu

Fengxiang He  
JD Explore Academy  
Beijing, Beijing, China  
fengxiang.f.he@gmail.com

Na Zou  
Texas A&M University  
College Station, TX, USA  
nzou1@tamu.edu

Dacheng Tao  
The University of Svdnev

Xia Hu  
Rice University

# Let's step back...

Transformers are not the right models for multiplication?  
Instead, Toolformers (Schick et. al. 2003)?

That's exactly the point!

Relatedly, are transformers the right models for **other  
compositional aspects of commonsense / language?**

Multiplication (+ puzzles, algorithms) are an “edge case”??? all other compositionality will work well with transformers + RLHF + scratchpad ???

1. How do we know **the full mastery?**
2. **WHY** is simple multiplication harder than other (seemingly more complex) compositional tasks?

# 2050: An AI Odyssey

Prolog: what CVPR 2050 be like

Chapter 1: The Possible Impossibilities

Chapter 2: The Impossible Possibilities

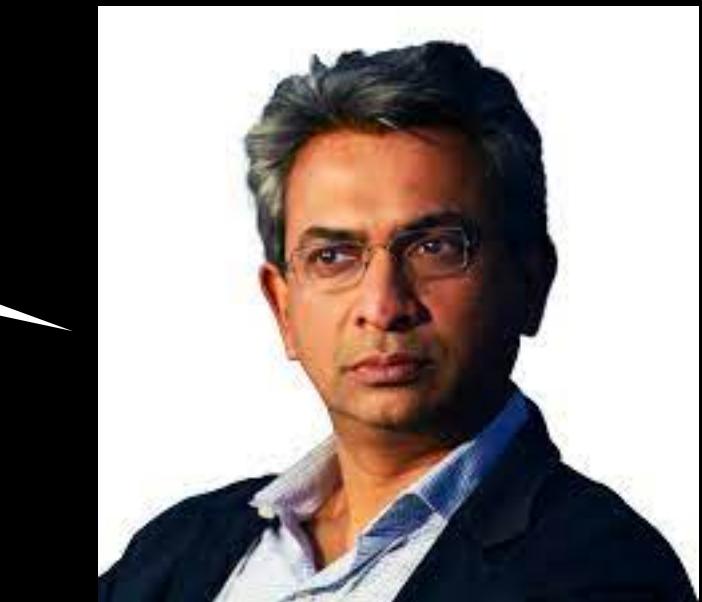
Chapter 3: The Paradox



# Circa 2023 ...

How can Indian startups create foundation models for India?

Rajan Anandan



Sam Atman

It's hopeless to compete with OpenAI





# Impossible Distillation

from Low-quality Model to High-Quality Dataset & Model  
for Summarization and Paraphrasing

— arxiv:2305.16635 —

Jaehun Jung



Peter West



Liwei Jiang



Faeze Brahman



Ximing Lu



Jillian Fisher



Taylor Sorensen

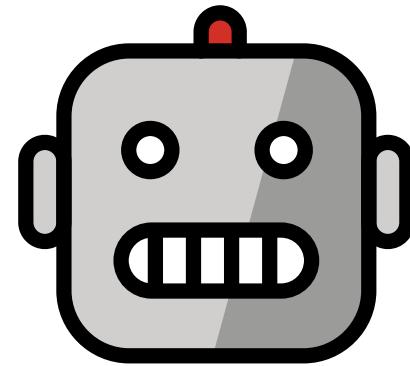


Yejin Choi



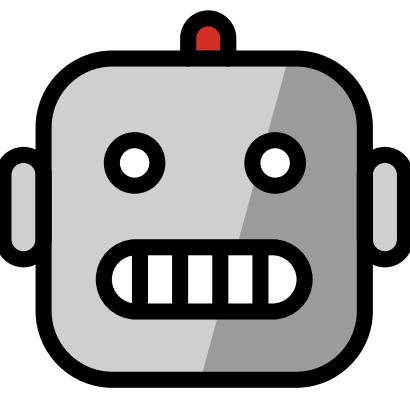
winning recipe = extreme-scale pre-training + RLHF at scale

GPT-2

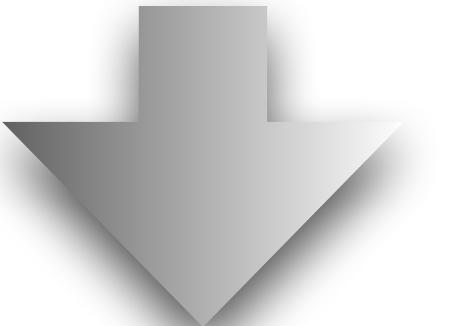
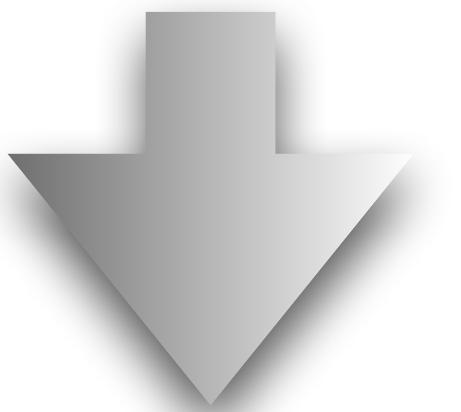


Low-quality, small models

???

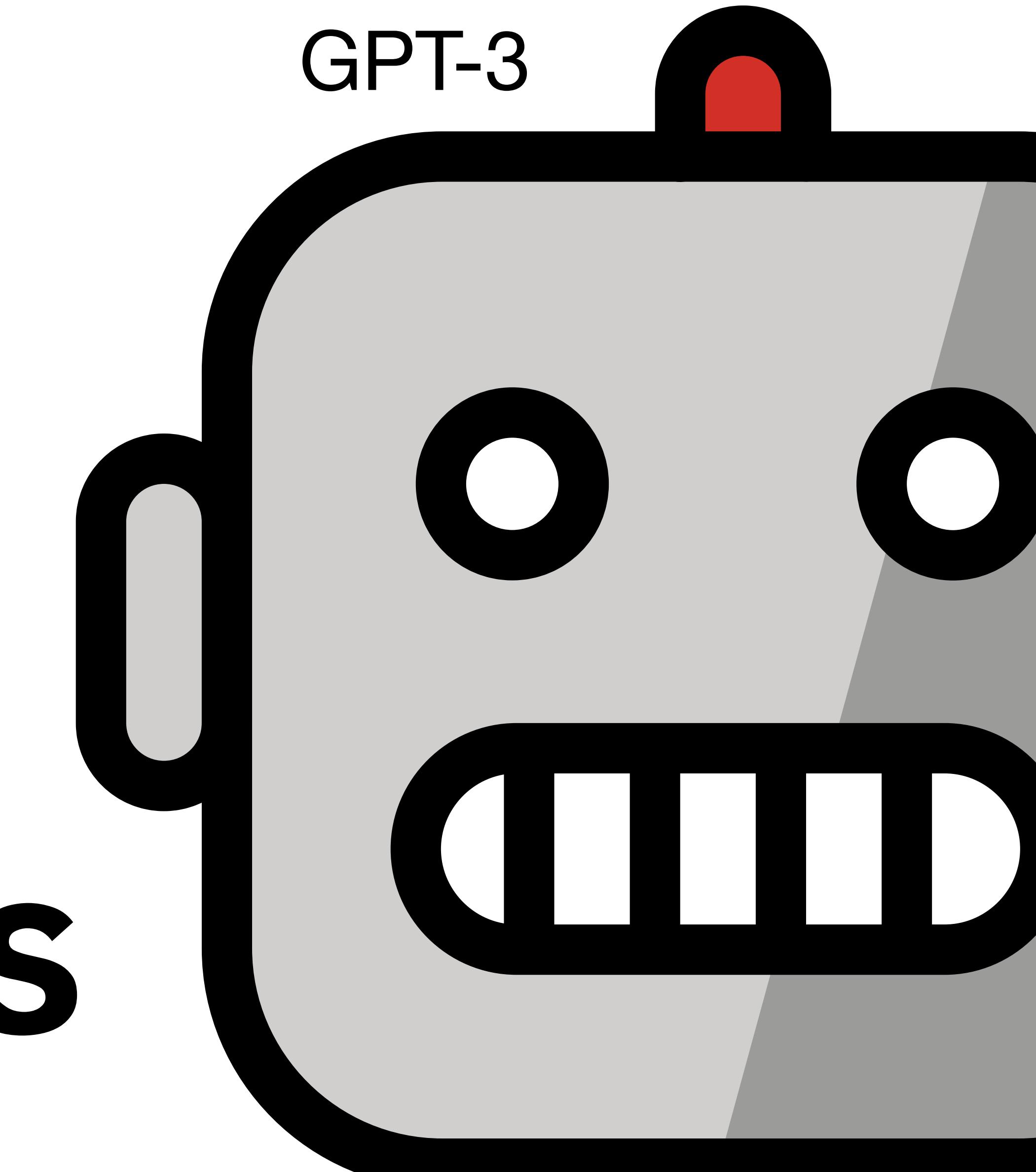


high-quality, small models



GPT-3

vs



# How is that even possible when imitating from proprietary LLMs are supposedly hopeless?

---

## The False Promise of Imitating Proprietary LLMs

---

**Arnav Gudibande\***  
UC Berkeley  
[arnavg@berkeley.edu](mailto:arnavg@berkeley.edu)

**Eric Wallace\***  
UC Berkeley  
[ericwallace@berkeley.edu](mailto:ericwallace@berkeley.edu)

**Charlie Snell\***  
UC Berkeley  
[csnell22@berkeley.edu](mailto:csnell22@berkeley.edu)

**Xinyang Geng**  
UC Berkeley  
[young.geng@berkeley.edu](mailto:young.geng@berkeley.edu)

**Hao Liu**  
UC Berkeley  
[hao.liu@berkeley.edu](mailto:hao.liu@berkeley.edu)

**Pieter Abbeel**  
UC Berkeley  
[pabbeel@berkeley.edu](mailto:pabbeel@berkeley.edu)

**Sergey Levine**  
UC Berkeley  
[svlevine@berkeley.edu](mailto:svlevine@berkeley.edu)

**Dawn Song**  
UC Berkeley  
[dawnsong@berkeley.edu](mailto:dawnsong@berkeley.edu)

# Are small LMs completely out of league?

Can small, off-the-shelf LMs learn to abstract without task supervision?



# Task-specific Symbolic Knowledge Distillation works!

## Symbolic Knowledge Distillation: from General Language Models to Commonsense Models

Peter West<sup>†‡\*</sup> Chandra Bhagavatula<sup>‡</sup> Jack Hessel<sup>‡</sup> Jena D. Hwang<sup>‡</sup>

Diego Bras<sup>‡</sup> Ximing Lu<sup>†‡</sup> Sean Welleck<sup>†‡</sup> Yejin Choi<sup>†‡\*</sup>

Computer Science & Engineering, University of Washington

and Institute for Artificial Intelligence

### Teaching Small Language Models to Reason

Lucie Charlotte Magister<sup>\*</sup>

University of Cambridge

lcm67@cam.ac.uk

Jonathan Mallinson

Google Research

jonmall@google.com

Jakub Adamek

Google Research

enkait@google.com

Eric Malmi  
Google Research  
emalmi@google.com

Aliaksai Severyn  
Google Research  
severyn@google.com

### Specializing Smaller Language Models towards Multi-Step Reasoning

Yao Fu<sup>◆</sup> Hao Peng<sup>◆</sup> Litu Ou<sup>◆</sup> Ashish Sabharwal<sup>◆</sup> Tushar Khot<sup>◆</sup>

### Distilling Step-by-Step! Outperforming Larger Language Models with Less Training Data and Smaller Model Sizes

Cheng-Yu Hsieh<sup>1,\*</sup>, Chun-Liang Li<sup>2</sup>, Chih-Kuan Yeh<sup>3</sup>, Hootan Nakhost<sup>2</sup>,  
Yasuhisa Fujii<sup>3</sup>, Alexander Ratner<sup>1</sup>, Ranjay Krishna<sup>1</sup>, Chen-Yu Lee<sup>2</sup>, Tomas Pfister<sup>2</sup>

<sup>1</sup>University of Washington, <sup>2</sup>Google Cloud AI Research, <sup>3</sup>Google Research

cydhsieh@cs.washington.edu

Our task in focus: learning to “**abstract**” in language

In NLP: ~ “sentence **summarization**”

✨ New observation: “**paraphrasing**” can be viewed as a special case of “**summarization**” ✨

# Misión Imposible:

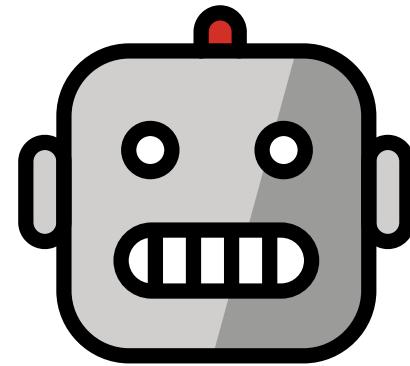
🔥 Learn to “summarize” + “paraphrase” 🔥

- without extreme-scale pre-training
- without RL with human feedback at scale
- without supervised datasets at scale

AI is as good as the data it was trained on

winning recipe = extreme-scale pre-training + RLHF at scale

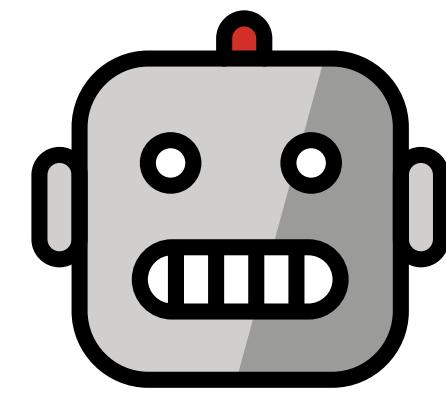
GPT-2



Low-quality, small models

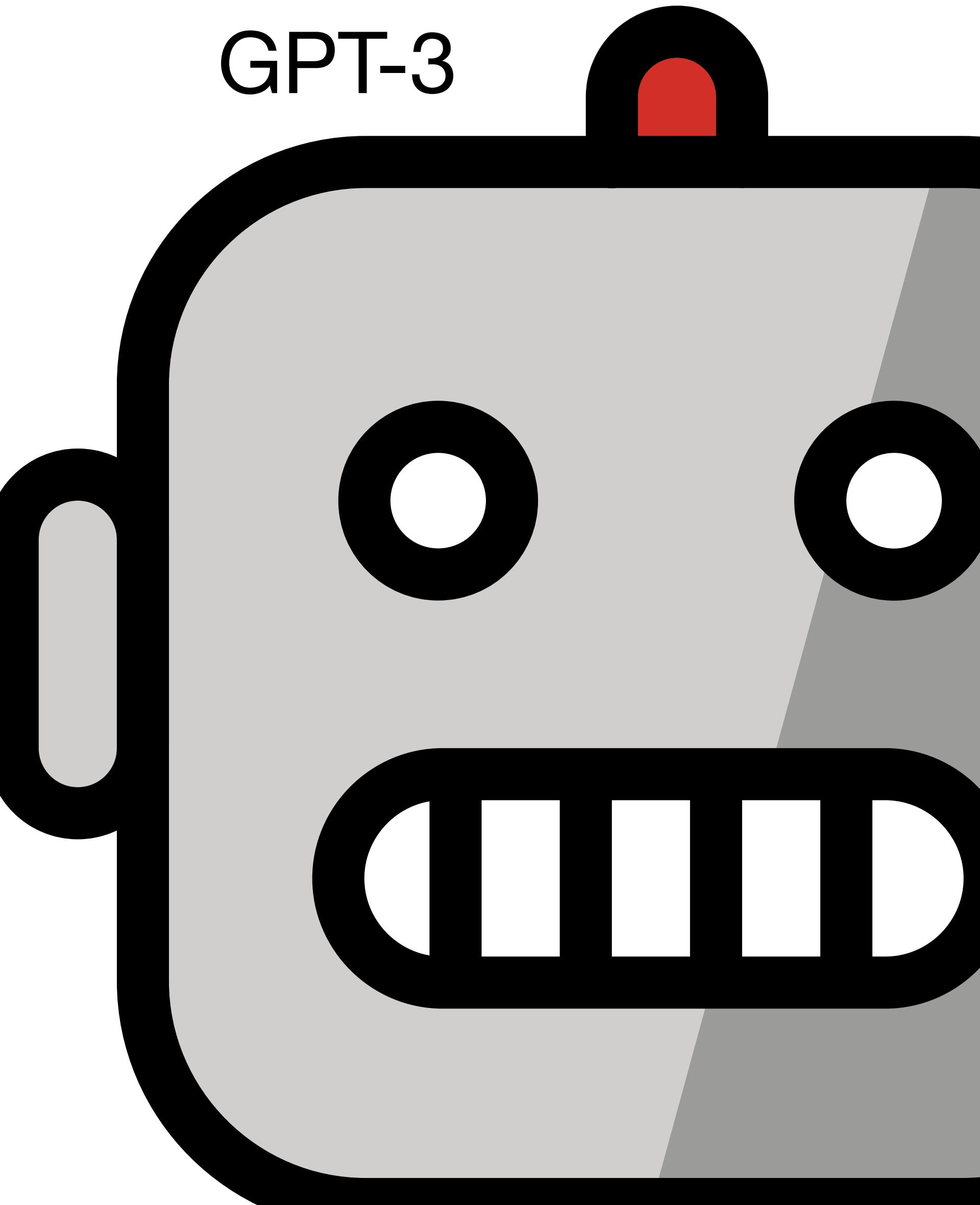


high-quality, large datasets



high-quality, small models

GPT-3



vs

# We will build on ...

## Symbolic Knowledge Distillation

From General Language Models to Commonsense Models

— NAACL 2022 —



Peter  
West

New:

*ATOMIC-10x*  
*COMET-distill*

Chandra  
Bhagavatula



Jack  
Hessel



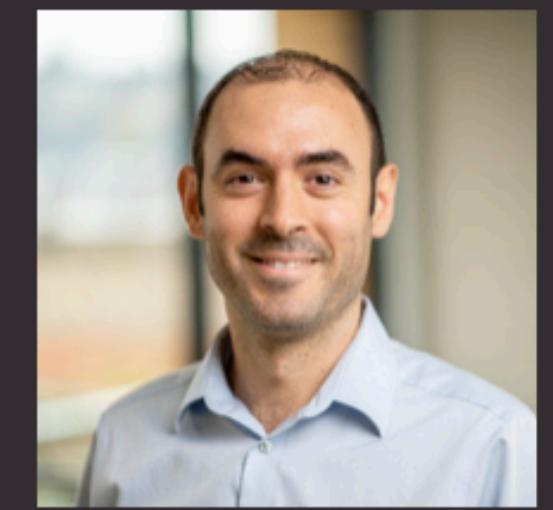
Jena  
Hwang



Liwei  
Jiang



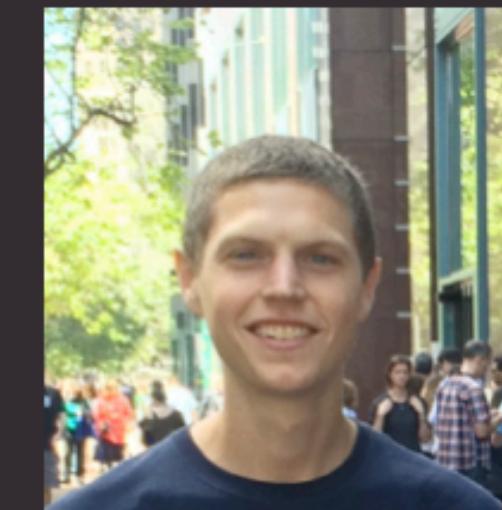
Ronan  
Le Bras



Ximing  
Lu



Sean  
Welleck



Yejin  
Choi



# Symbolic Knowledge Distillation

From General Language Models to Commonsense Models

— NAACL 2022 —



Peter  
West

Chandra  
Bhagavatula



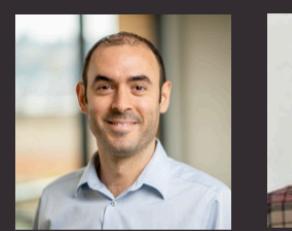
Jack  
Hessel



Jena  
Hwang



Liwei  
Jiang



Ronan  
Le Bras



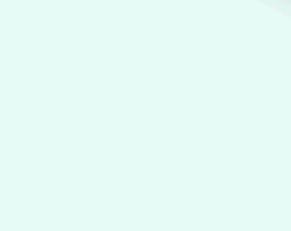
Ximing  
Lu



Sean  
Welleck



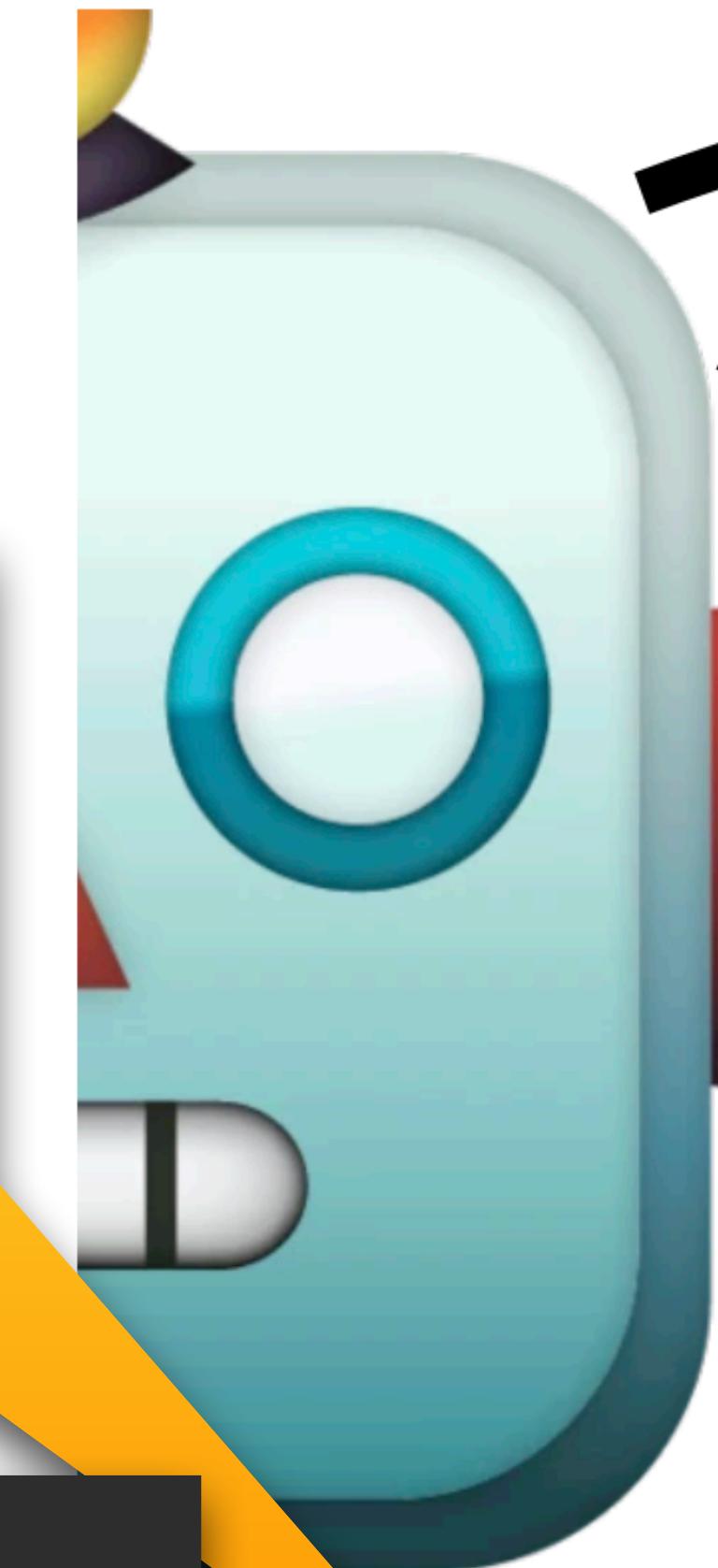
Yejin  
Choi



ATOMIC-10x: a machine-authored KB that wins, for the first time, over a human-authored KB in all criteria: scale, accuracy, and diversity.



Yeah but can we get anywhere without GPT-3?



GPT-3

175B Parameters  
General Model

Symbolic Knowledge  
Distillation

CRITIC

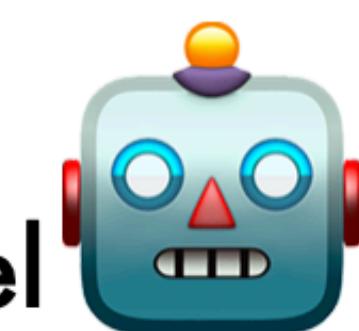
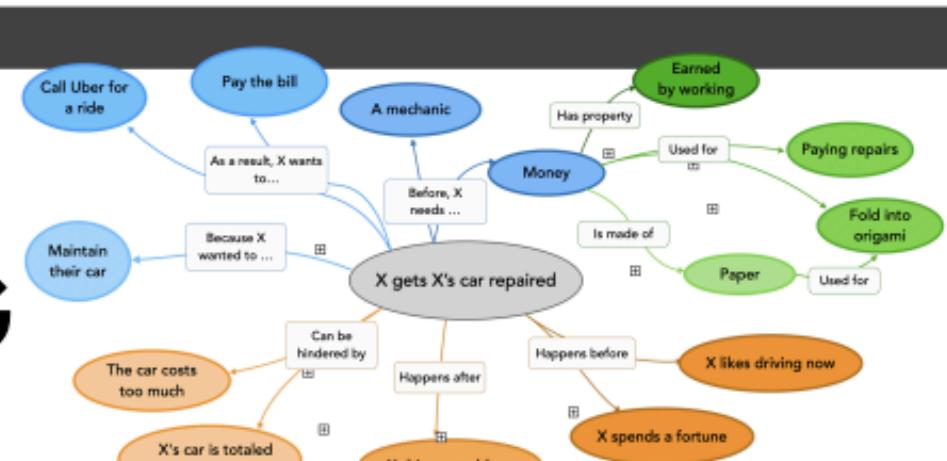
Fine-tuned RoBERTa  
filters for quality

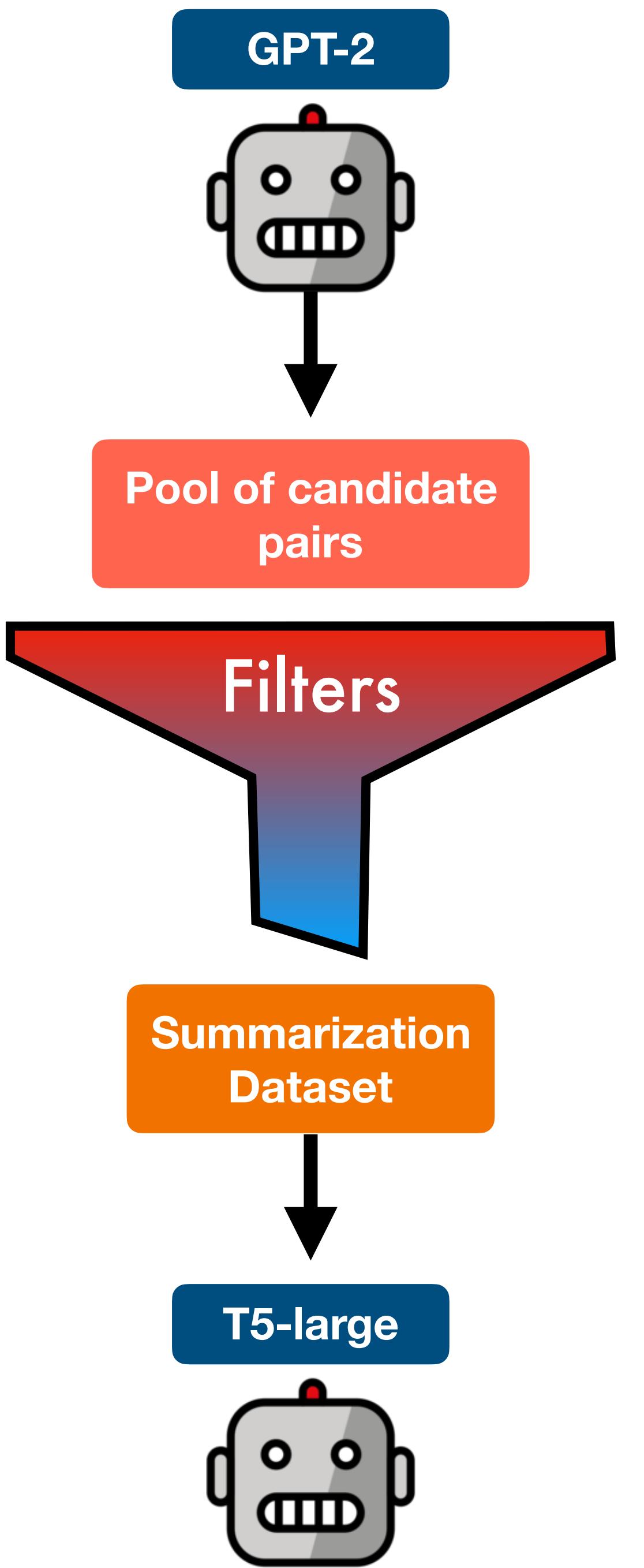
ATOMIC<sup>10X</sup>

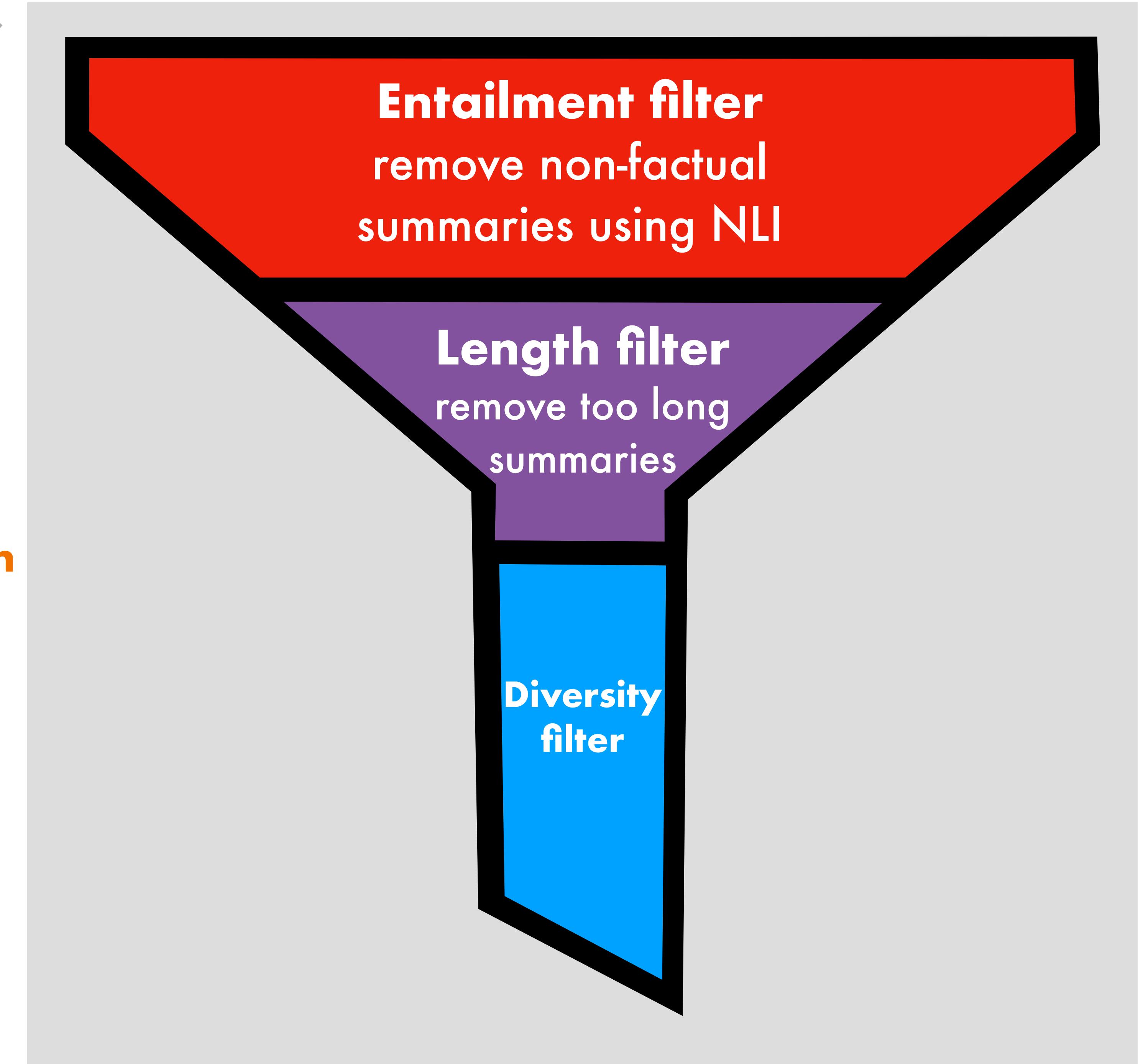
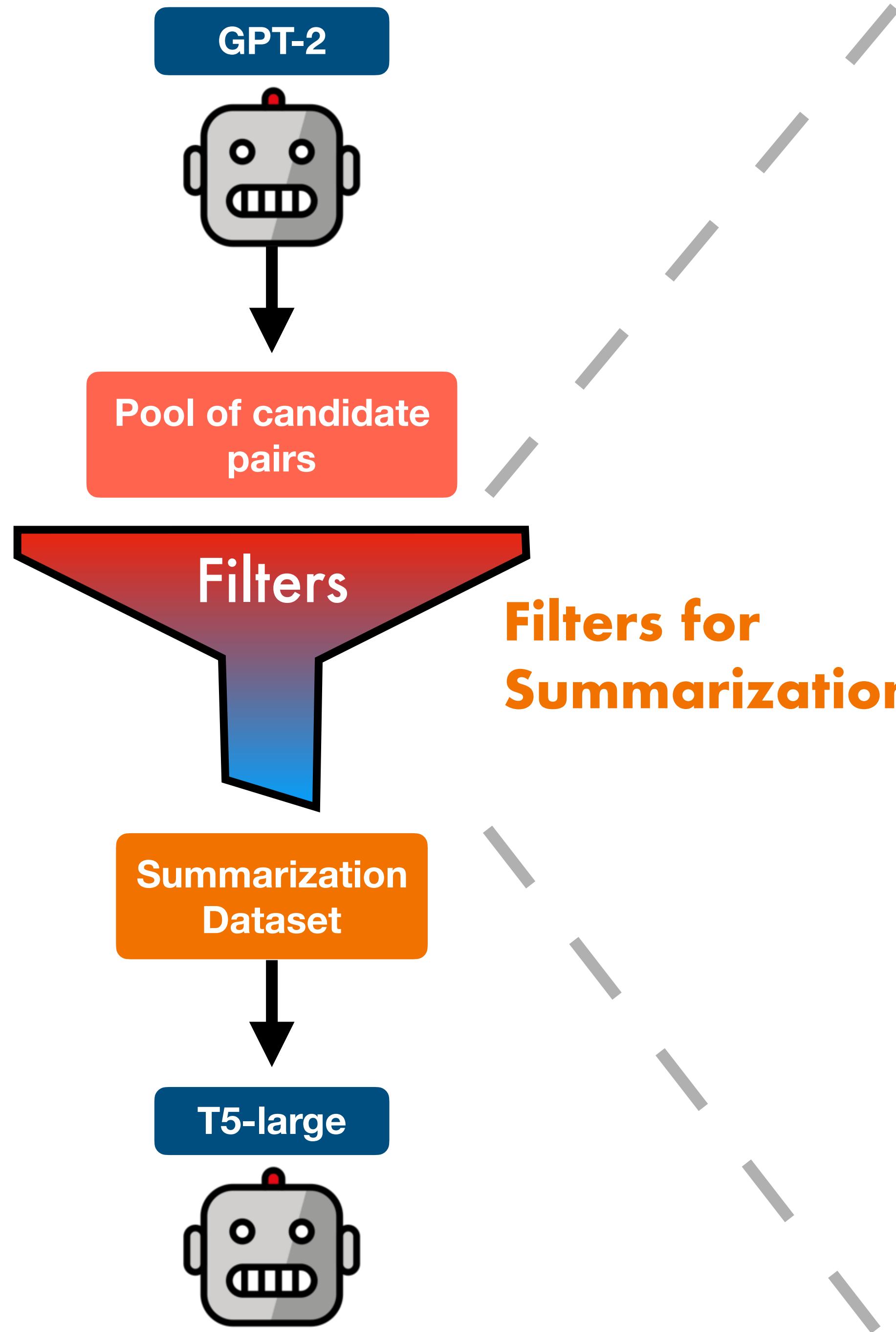
6.5M Examples  
Commonsense KG

COMET<sup>distil</sup>

1.5B Parameters  
Commonsense Model







**Bidirectional  
Entailment filter**

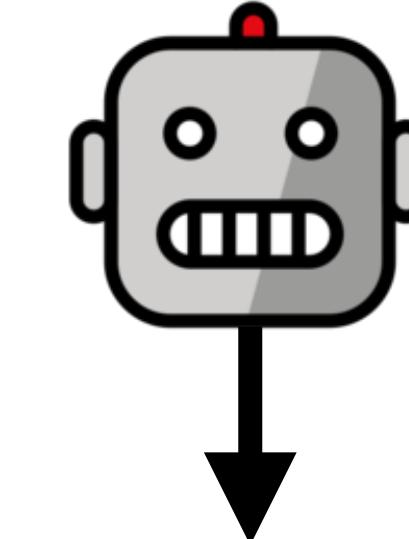
**Abstractness  
filter**

**Length  
filter**

**Diversity  
filter**

**Filters for  
Paraphrasing**

**GPT-2**

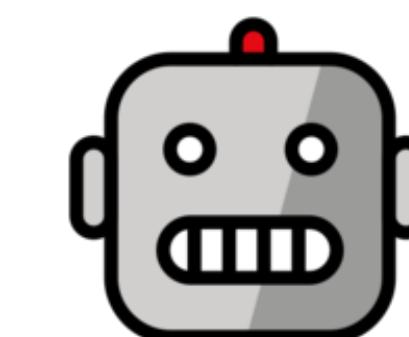


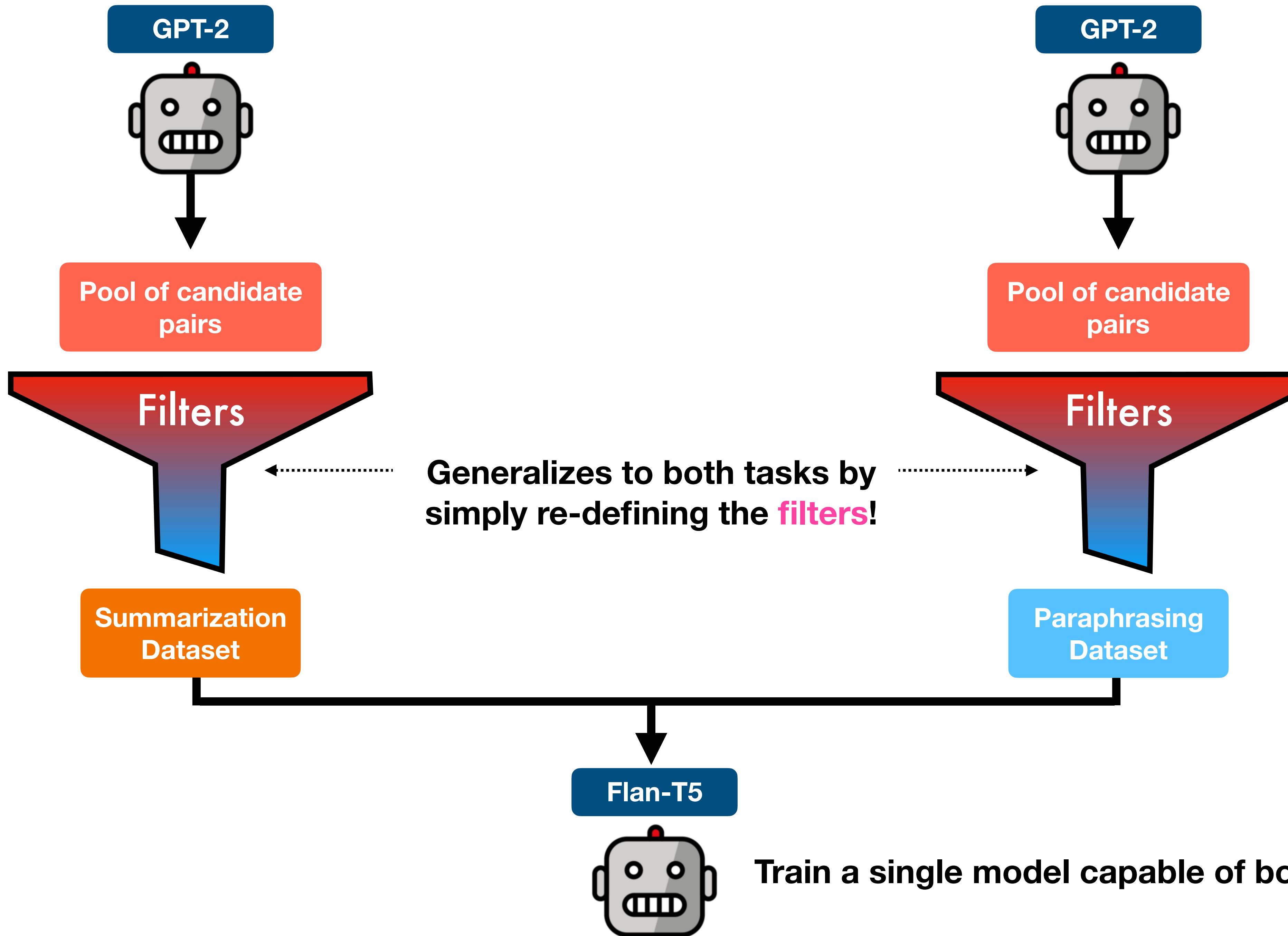
**Pool of candidate  
pairs**

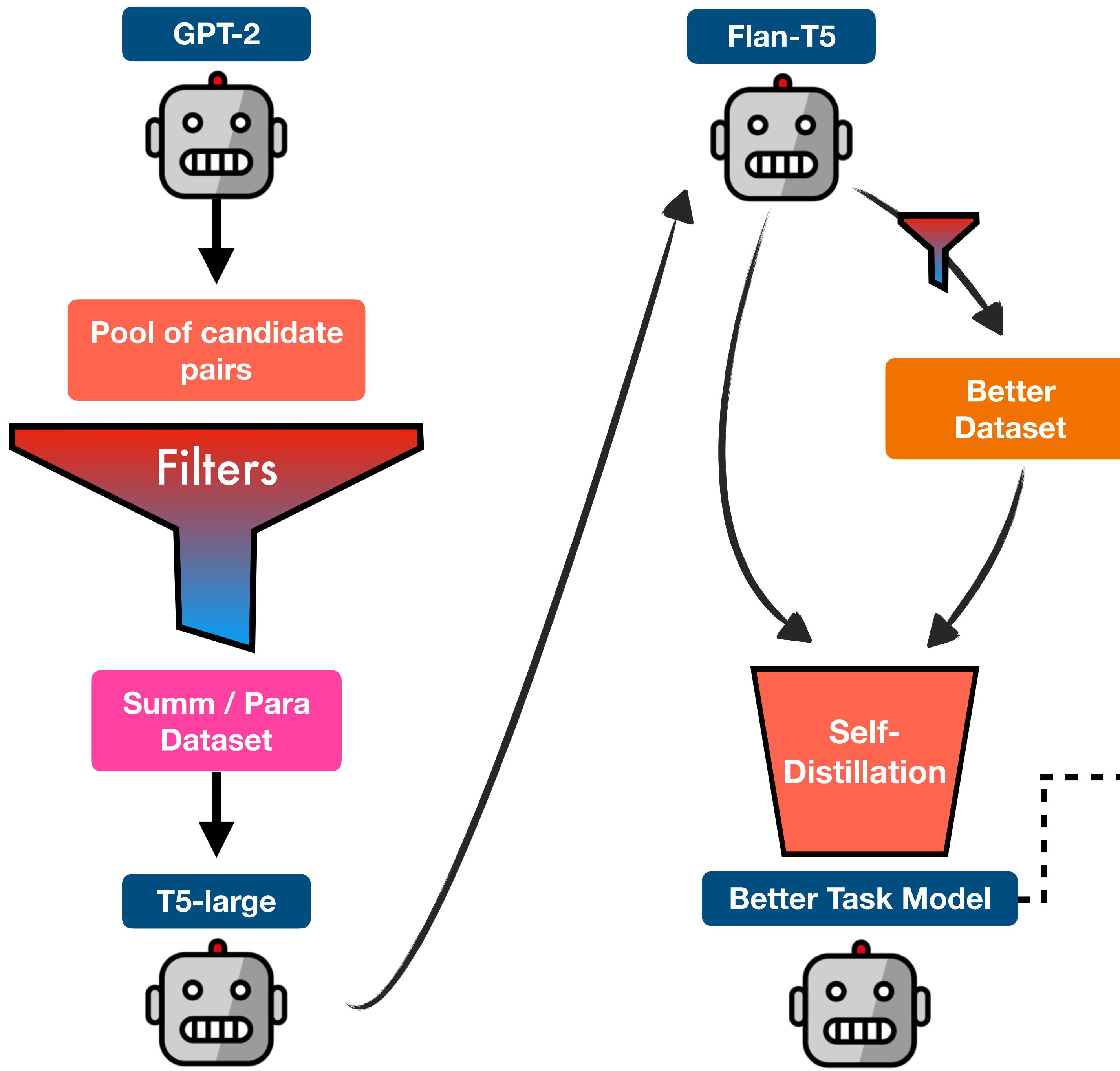
**Filters**

**Paraphrasing  
Dataset**

**T5-large**







**Self-Distillation**  
yields better dataset,  
stronger task model

DimSum+

3.4M samples for  
sentence summarization + paraphrasing,  
spanning news / reddit / bio domains

T5-ImpDistill

770M LM capable of both  
controllable summarization + paraphrasing,  
distilled purely from < 2B LMs

"While we will be looking across all parts of the newsroom, at the end of the redundancy program we expect there will be significantly fewer editorial management, video, presentation and section writer roles," the publisher is quoted as saying in an internal note.

🔥 T5-ImpDistill

"We are looking to reduce the number of staff in the newsroom", the publisher said in an internal note.



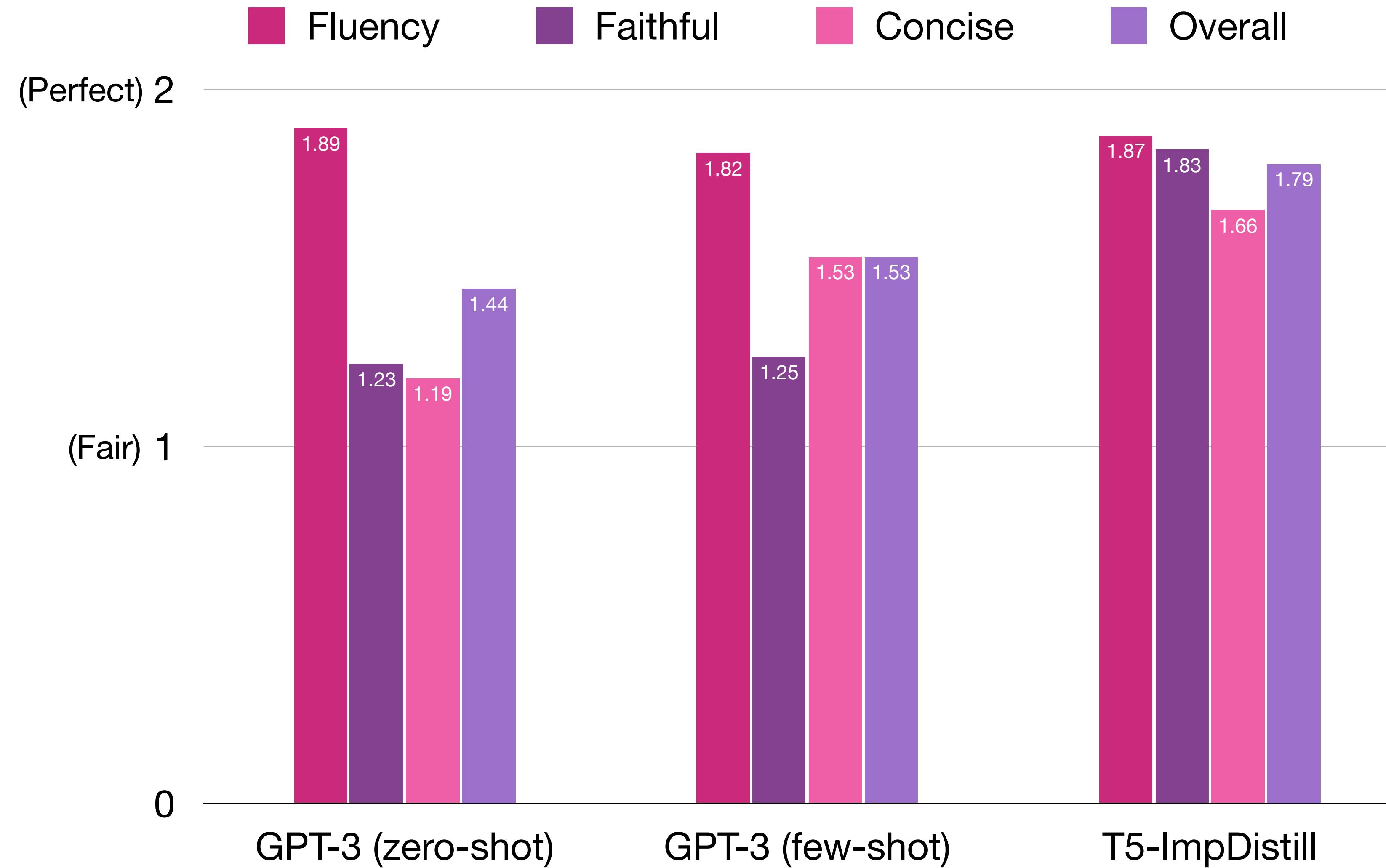
GPT-3 text-davinci-003, zero-shot

The publisher has informed staff through an internal note that, after implementing a redundancy program, there will be a significant reduction in the number of editorial management, video, presentation, and section writer roles.



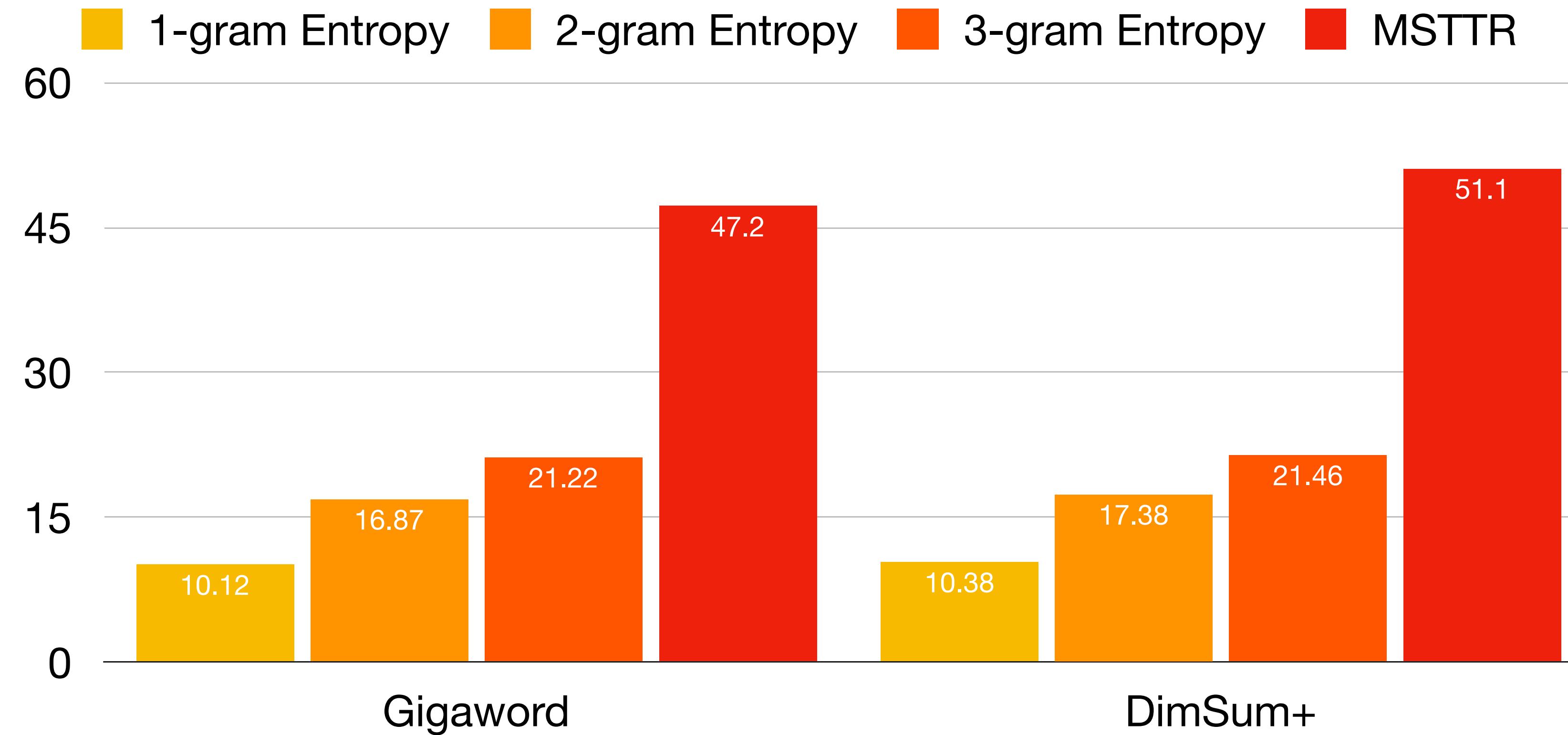
*hallucinating unsupported content*

# Stronger than 200x larger GPT-3 in human evaluation!



# Dataset has higher diversity than human-authored **Gigaword**

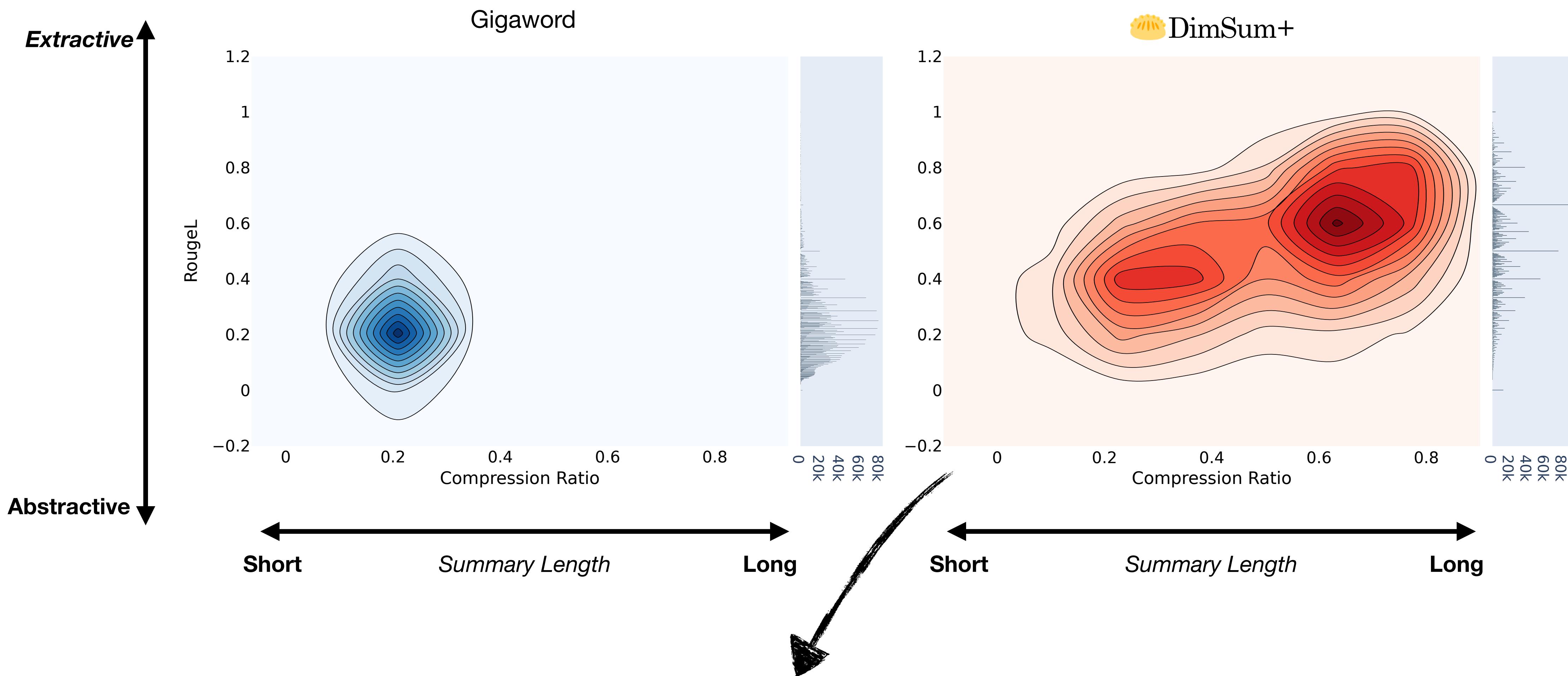
(Rush et al. 2015)



Our dataset (3.4M) exhibit more **lexical diversity** than human-authored Gigaword (4M)!

# Dataset has higher diversity than human-authored **Gigaword**

(Rush et al. 2015)



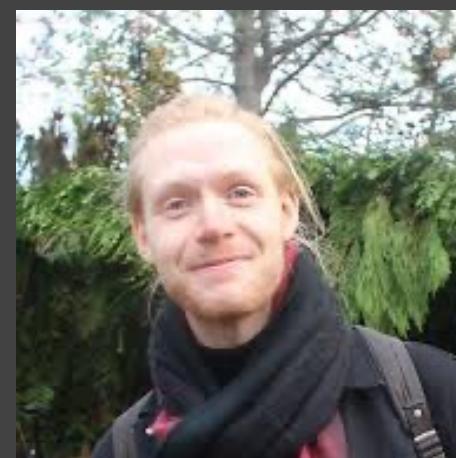
Our dataset covers **diverse summarization strategy!**

# NEUROLOGIC A⭐ Constrained Text Generation with Lookahead Heuristic

Sean Welleck



Peter West



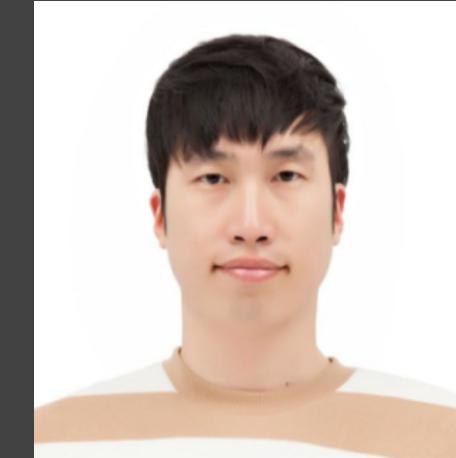
Liwei Jiang



Lianhui Qin



Youngjae Yu



Ximing Lu



Daniel Khashabi



Jungo Kasai



Ronan Le Bras



Rowan Zellers

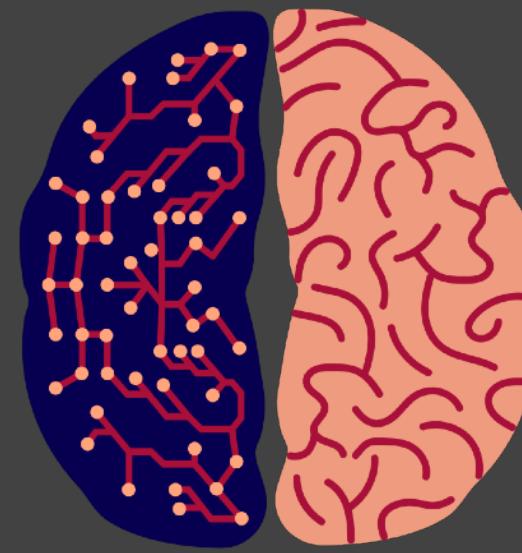


Noah Smith



Yejin Choi





# NEUROLOGIC DECODING

(Un)supervised Neural Text Generation with Predicate Logic Constraints

—NAACL 2021—

Ximing Lu



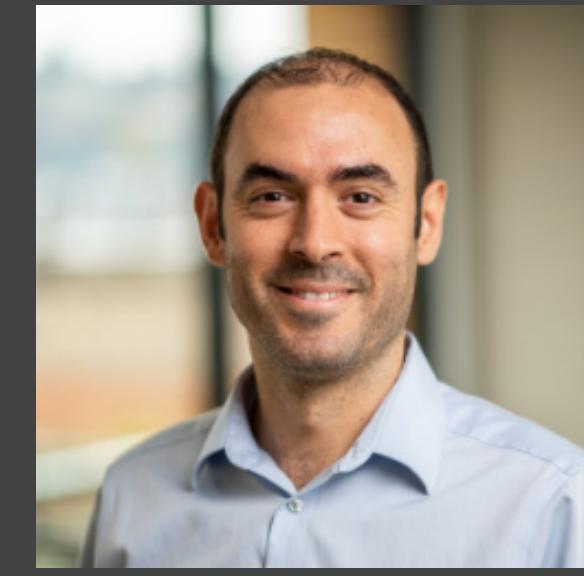
Peter  
West



Rowan  
Zellers



Ronan  
LeBras



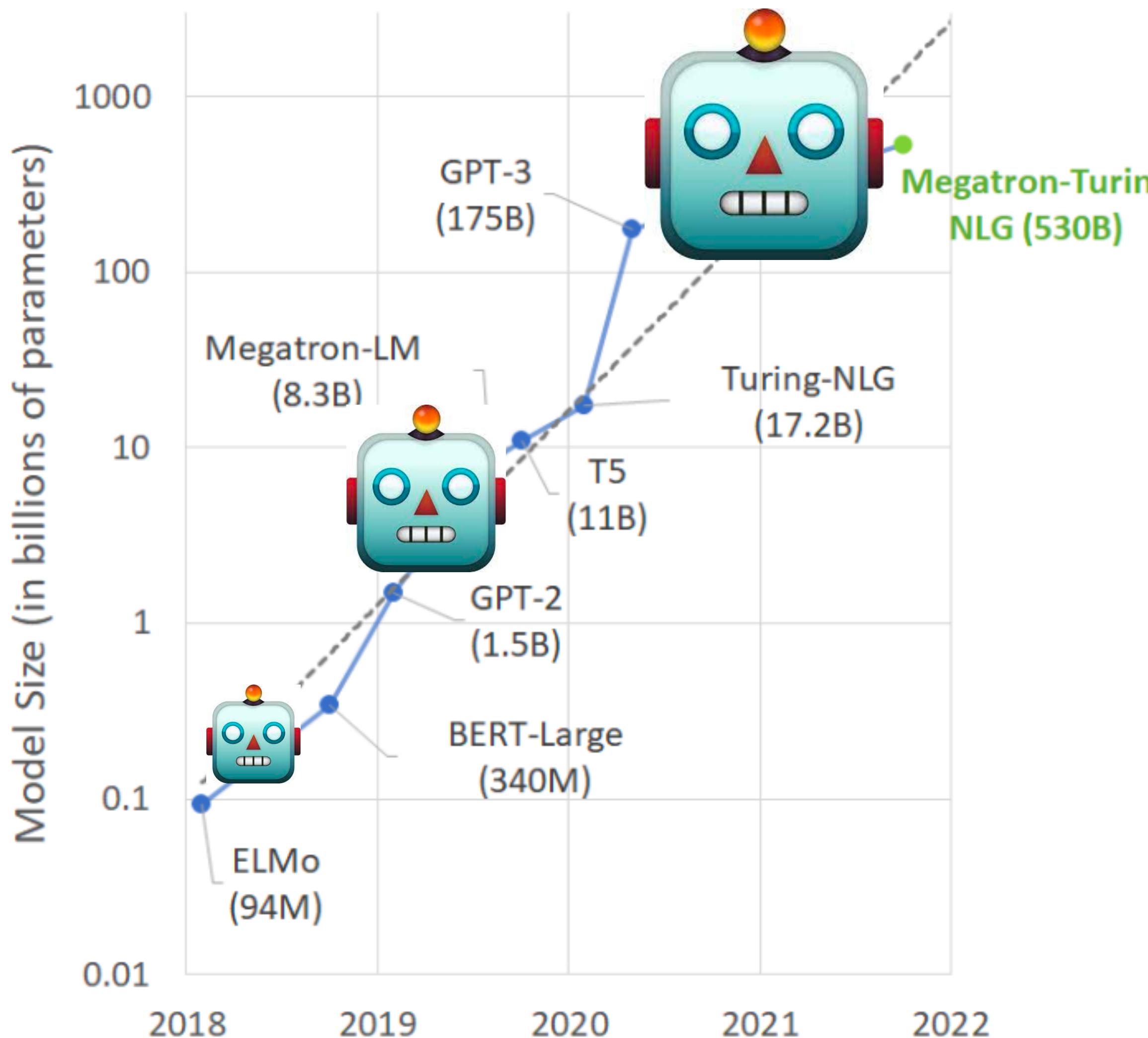
Chandra  
Bhagavatula



Yejin  
Choi



# Neural Language Models



## Program Synthesis

### Your AI pair programmer

With GitHub Copilot, get suggestions for whole lines or entire functions right inside your editor.

Sign up >

assert check(70) == P  
assert check(23) == P  
assert check(73) == T

def check(n):

if n == 2\*int(str(n)[::2]):

return True

else:

return False

GitHub Copilot

## Long-form QA

Low exponentially in the last 50 years?

175B best-of-64

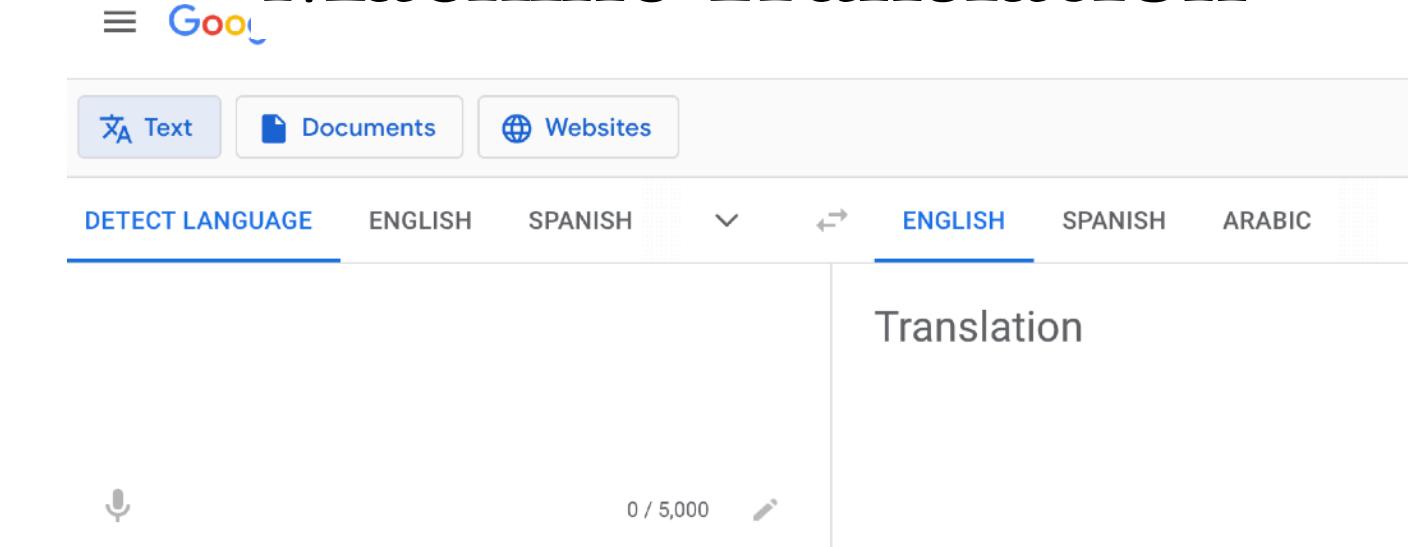
many explanations for the exponential growth in technology in tury. One explanation is that the pace of technological eeds up exponentially over time because of a common force arward<sup>[3]</sup>. Another explanation is that each new generation of stands on the shoulders of its predecessors, allowing for

## Open-Ended Generation

Build next-gen apps with OpenAI's powerful models.

OpenAI's API provides access to GPT-3, which performs a wide variety of natural language tasks, and Codex, which translates natural language to code.

## Machine Translation



## Dialogue

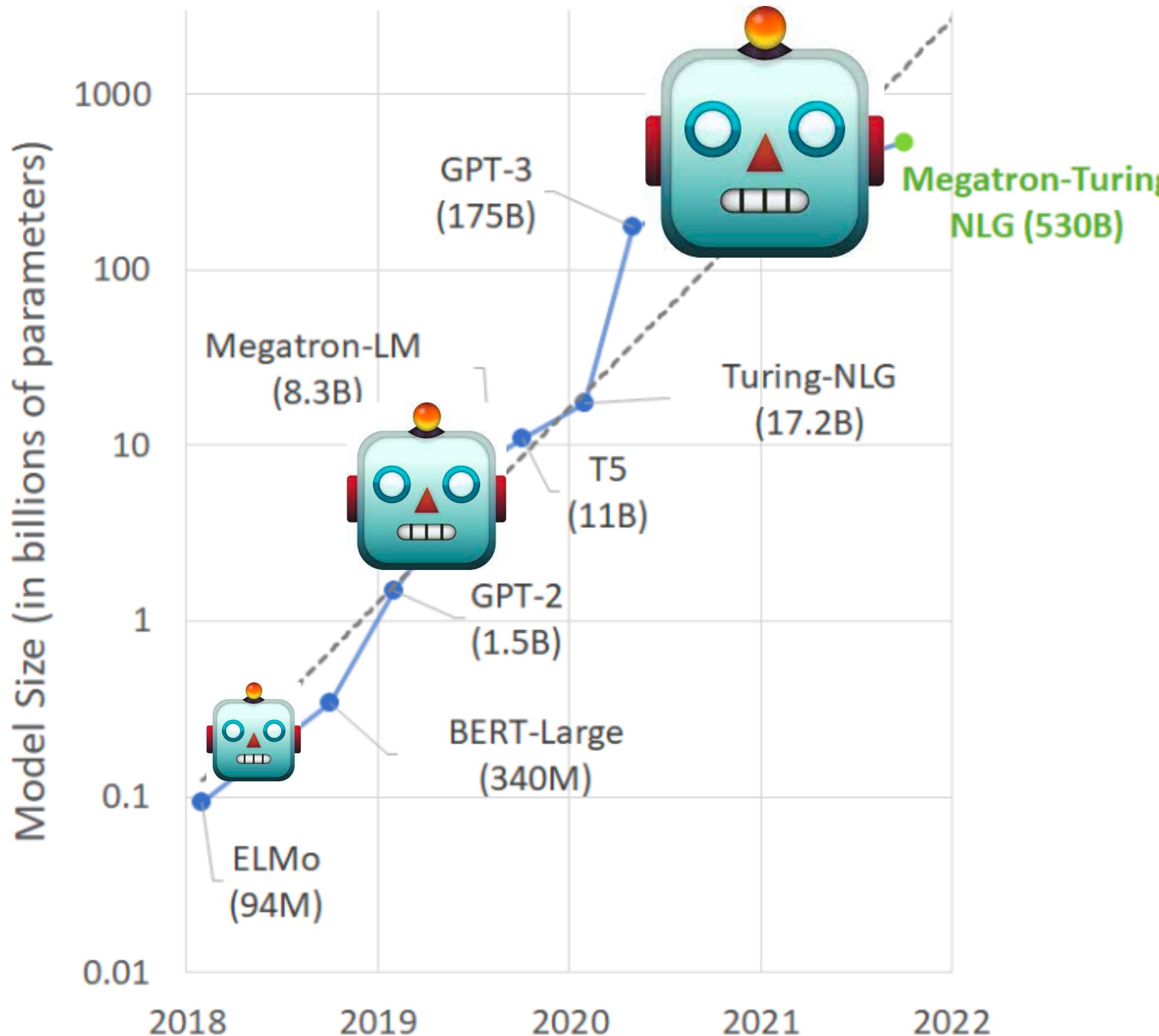
am a friendly dialog model. What do you want to talk about?

What's a good topic for a new blog?

are so many! How about something w food item that you just tried.

# Neural Language Models

COMMONGEN  
(Liu et al 2020)

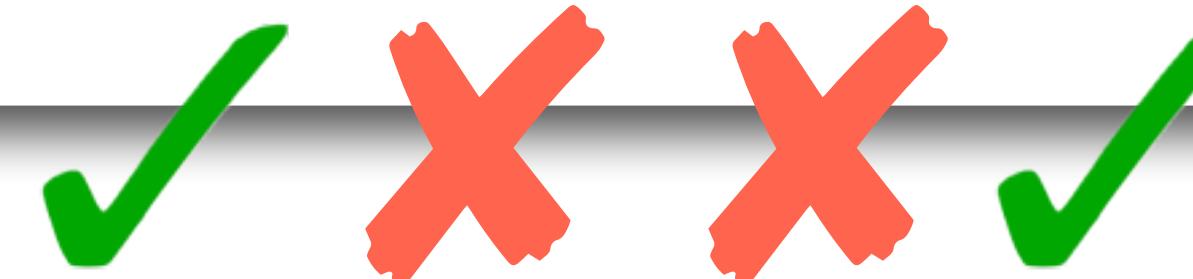


What is the [mass](#) of Jupiter?

Language Model  
(GPT3)

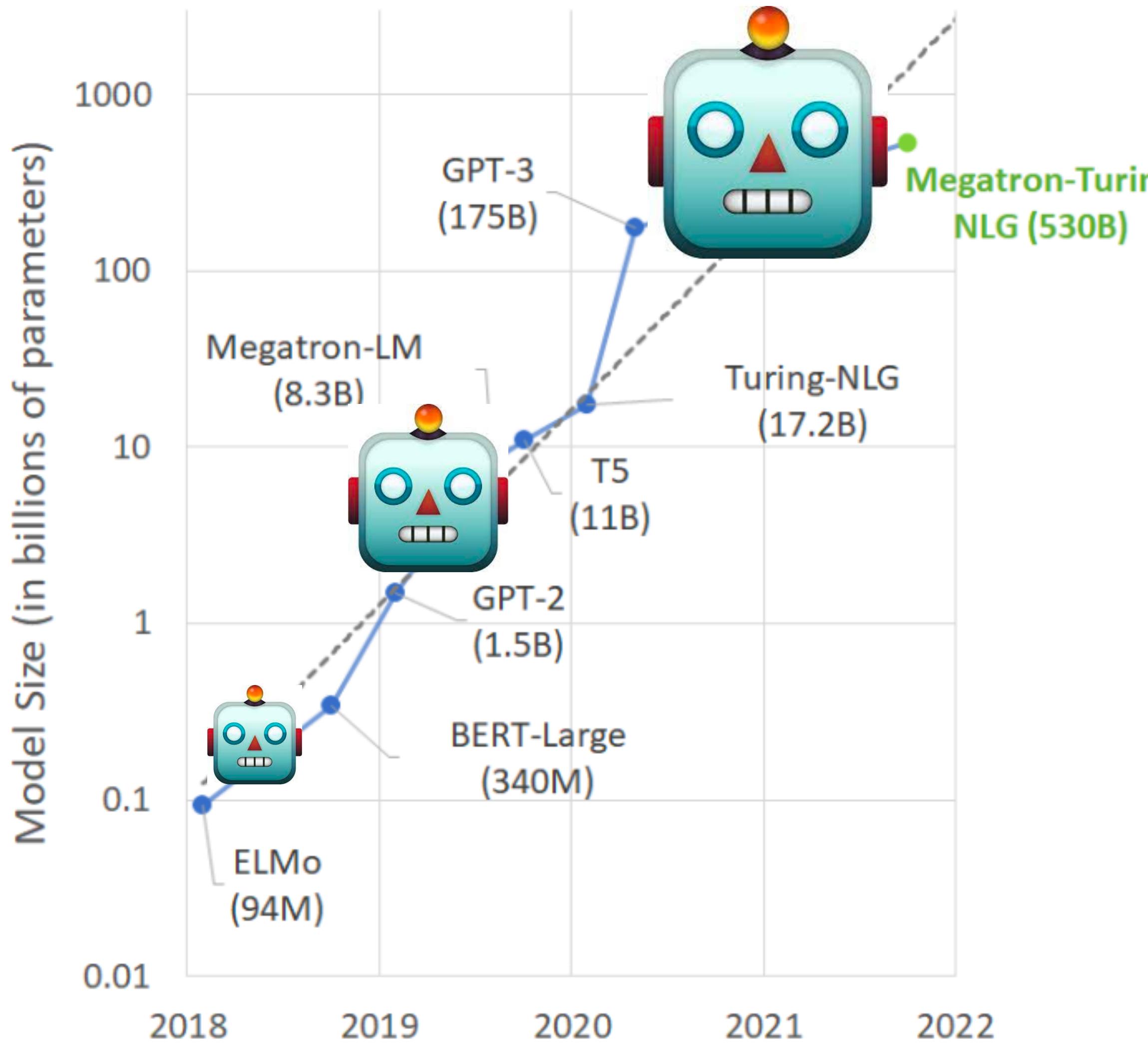
Generate a question containing all of the given words.

Words: [Jupiter](#), [Mercury](#), [Venus](#), [mass](#)

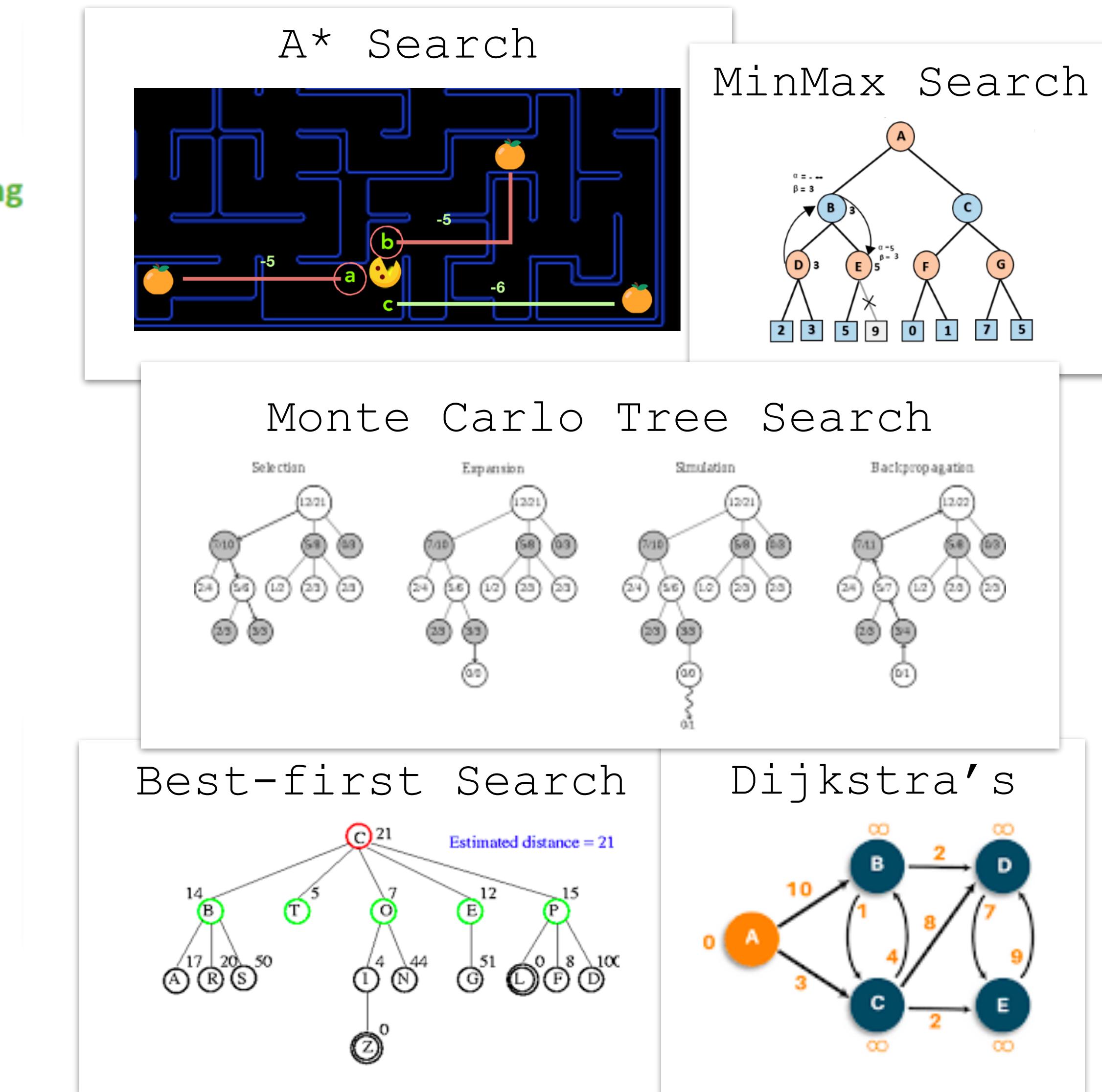


missing keywords

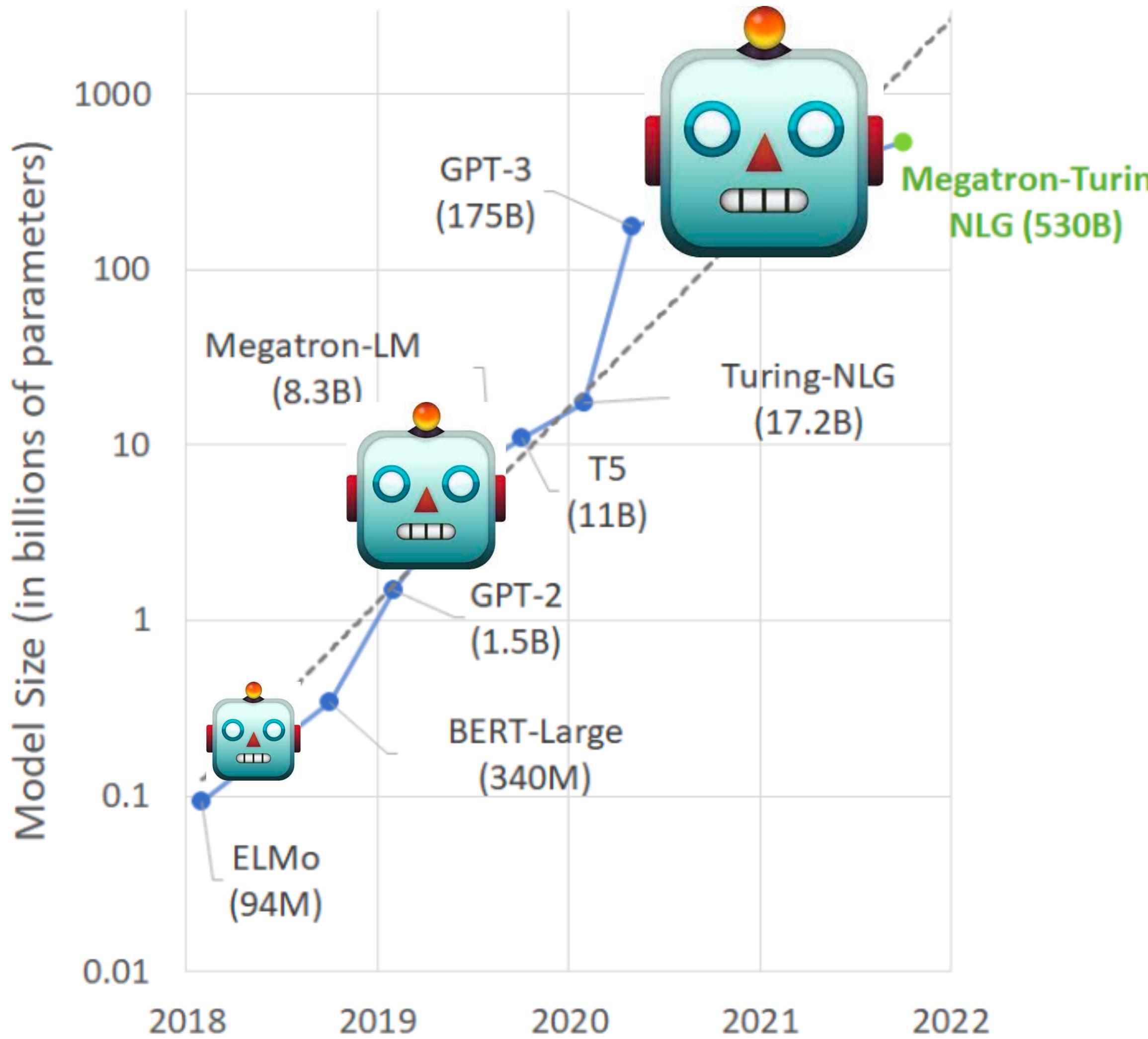
# Neural Language Models



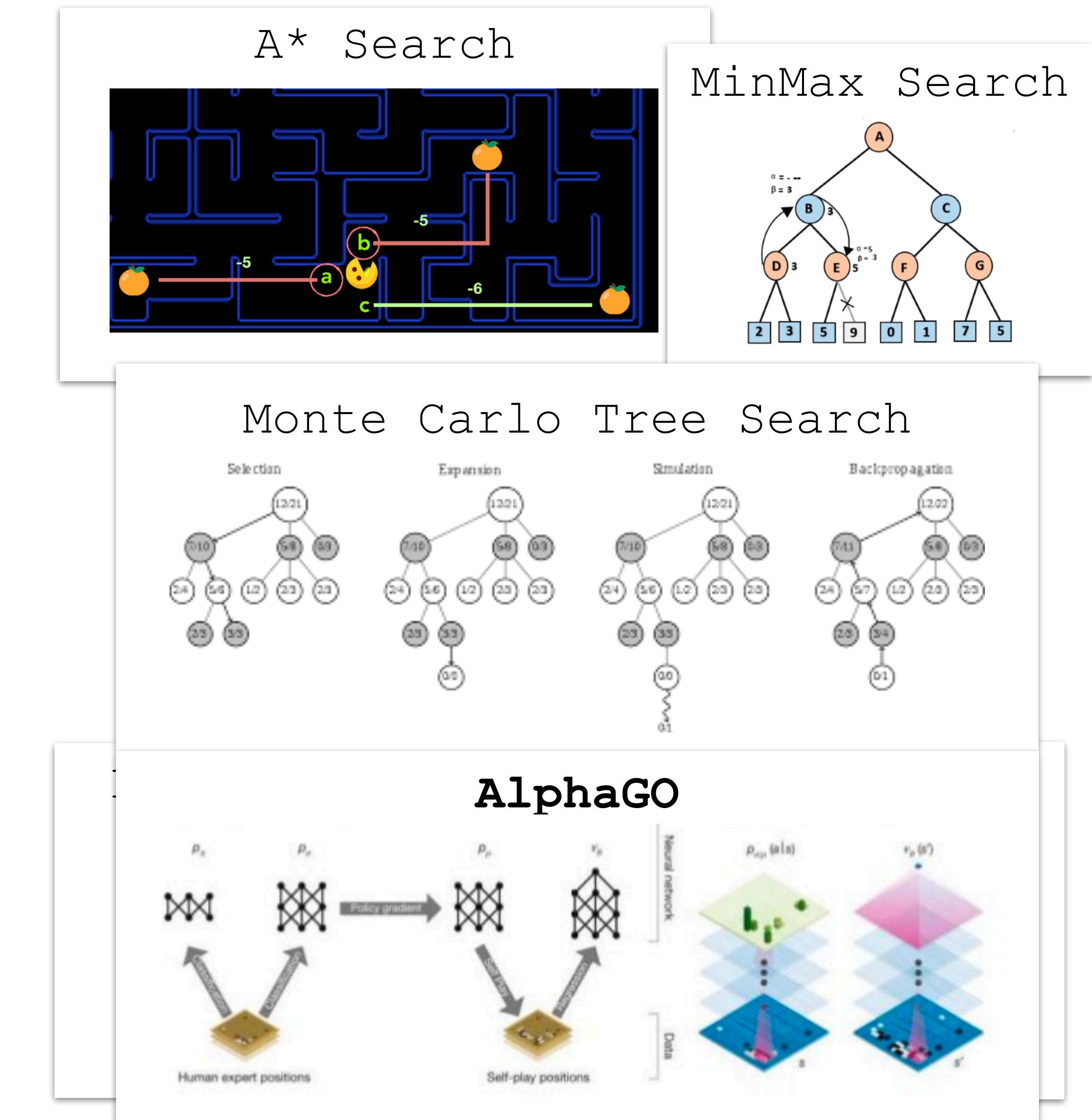
# Search Algorithms in Classical AI



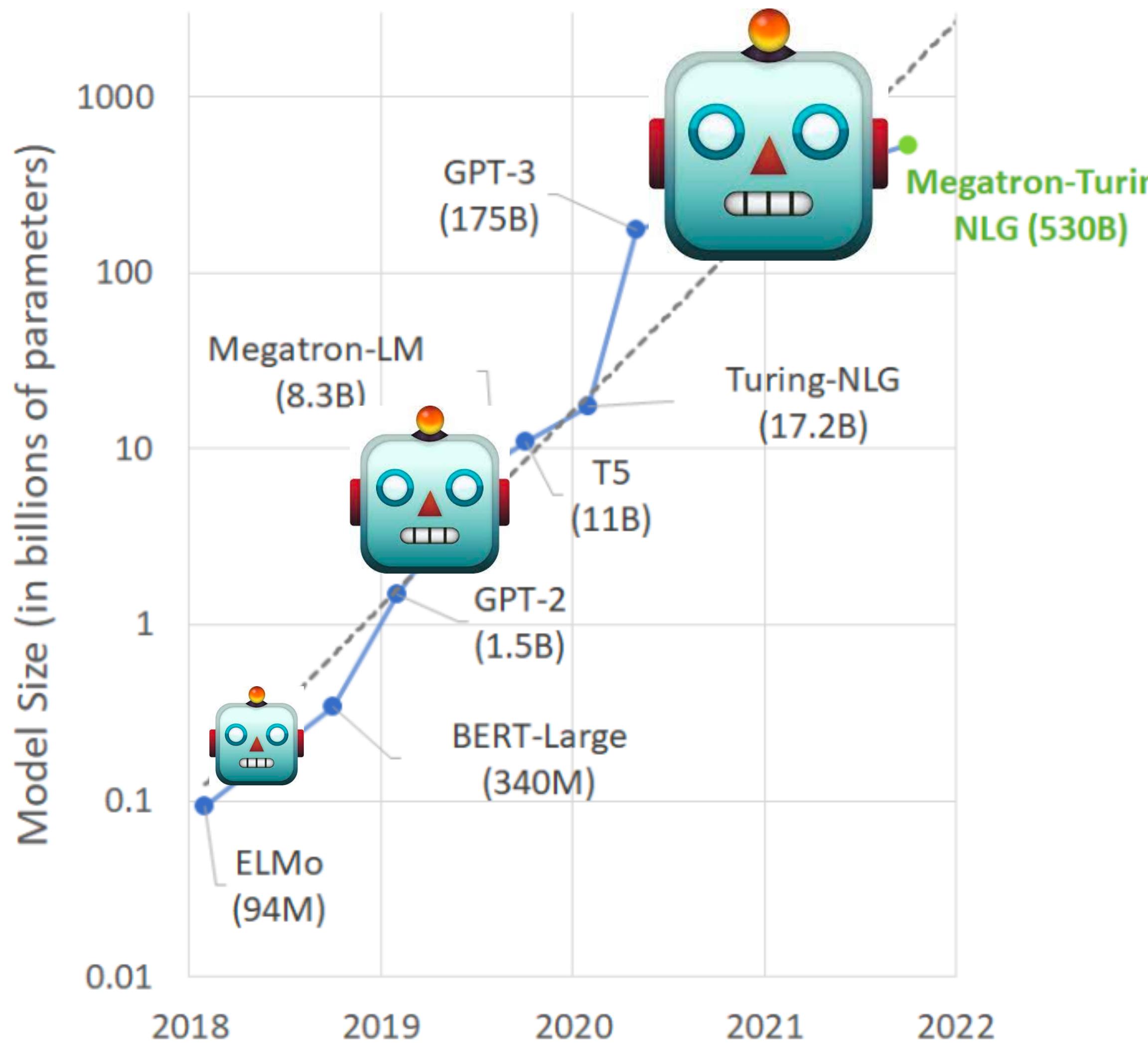
# Neural Language Models



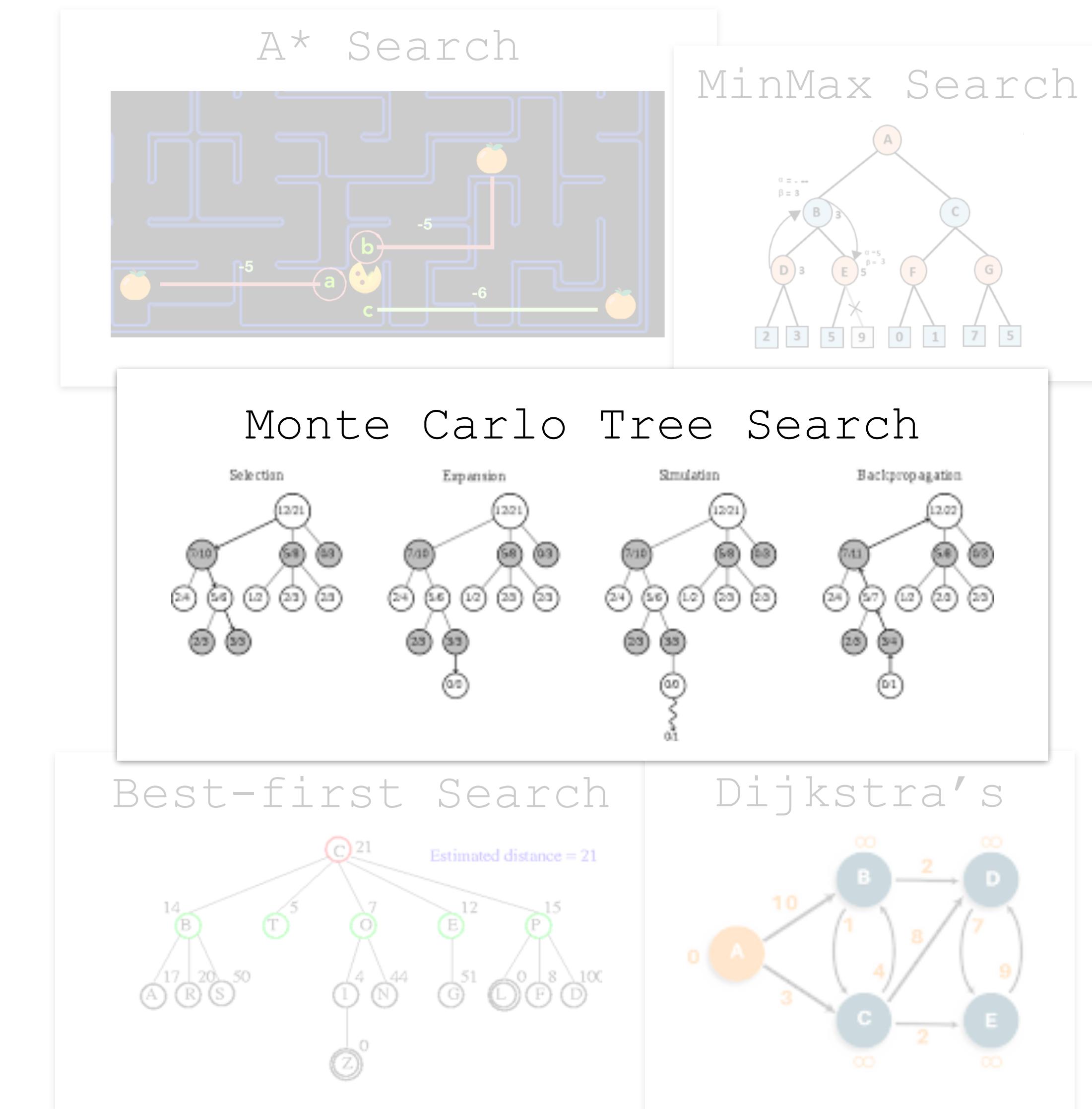
# Search Algorithms in Classical AI



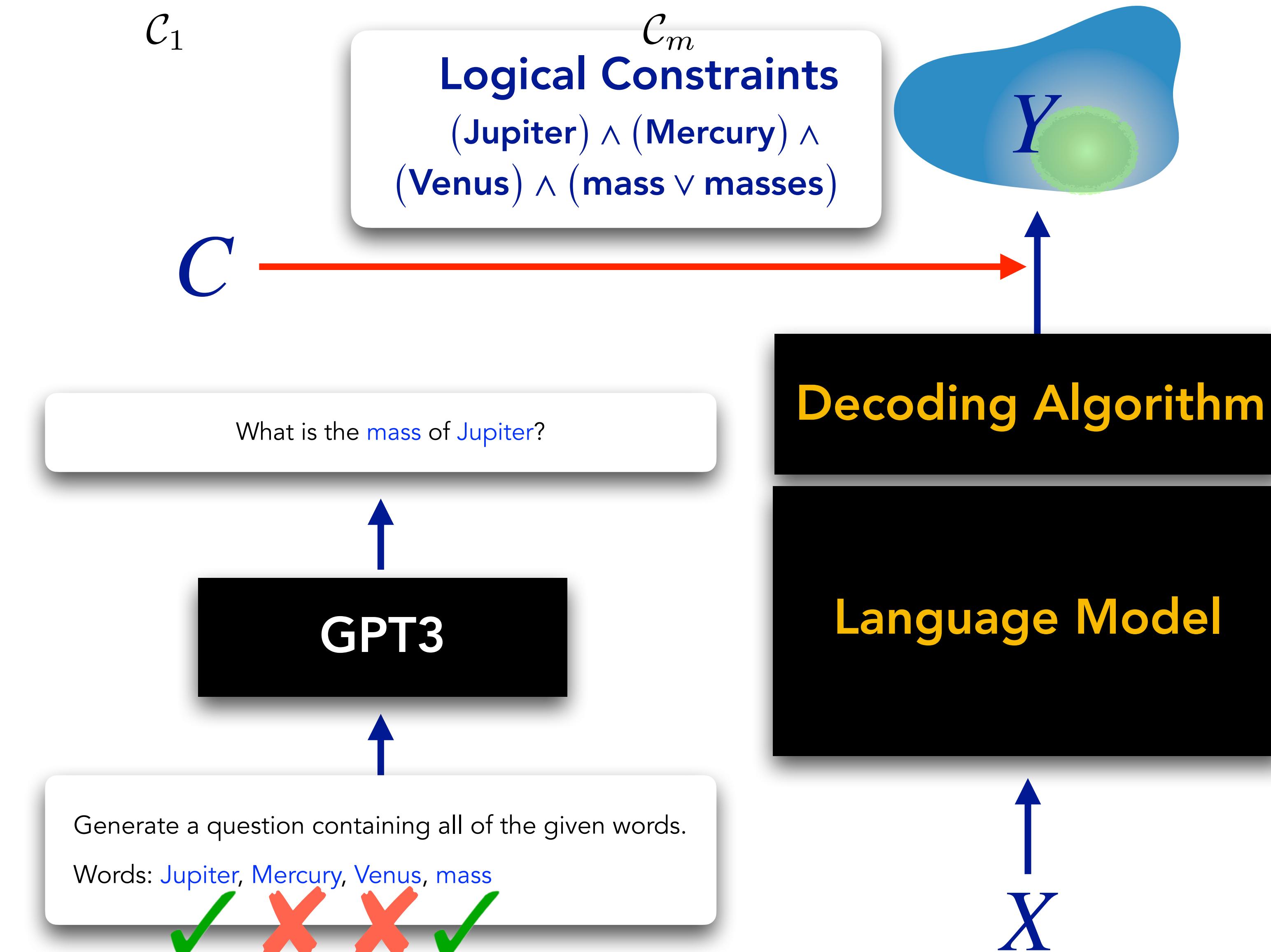
# Neural Language Models



# Search Algorithms in Classical AI



$$\underbrace{(\mathcal{D}_1 \vee \mathcal{D}_2 \dots \vee \mathcal{D}_i)}_{\mathcal{C}_1} \wedge \dots \wedge \underbrace{(\mathcal{D}_k \vee \mathcal{D}_{k+1} \dots \vee \mathcal{D}_l)}_{\mathcal{C}_m}$$

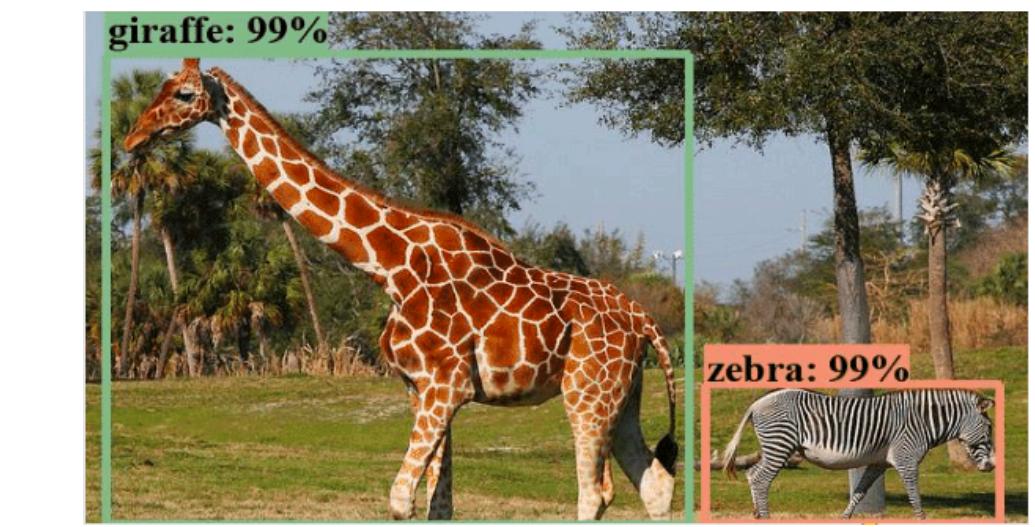


### Table to Text

$X$	type	hotel
$Y$	count	182
$Z$	dogs allowed	don't care

There are 182 hotels if you do not care whether dogs are allowed.

### Image Captioning

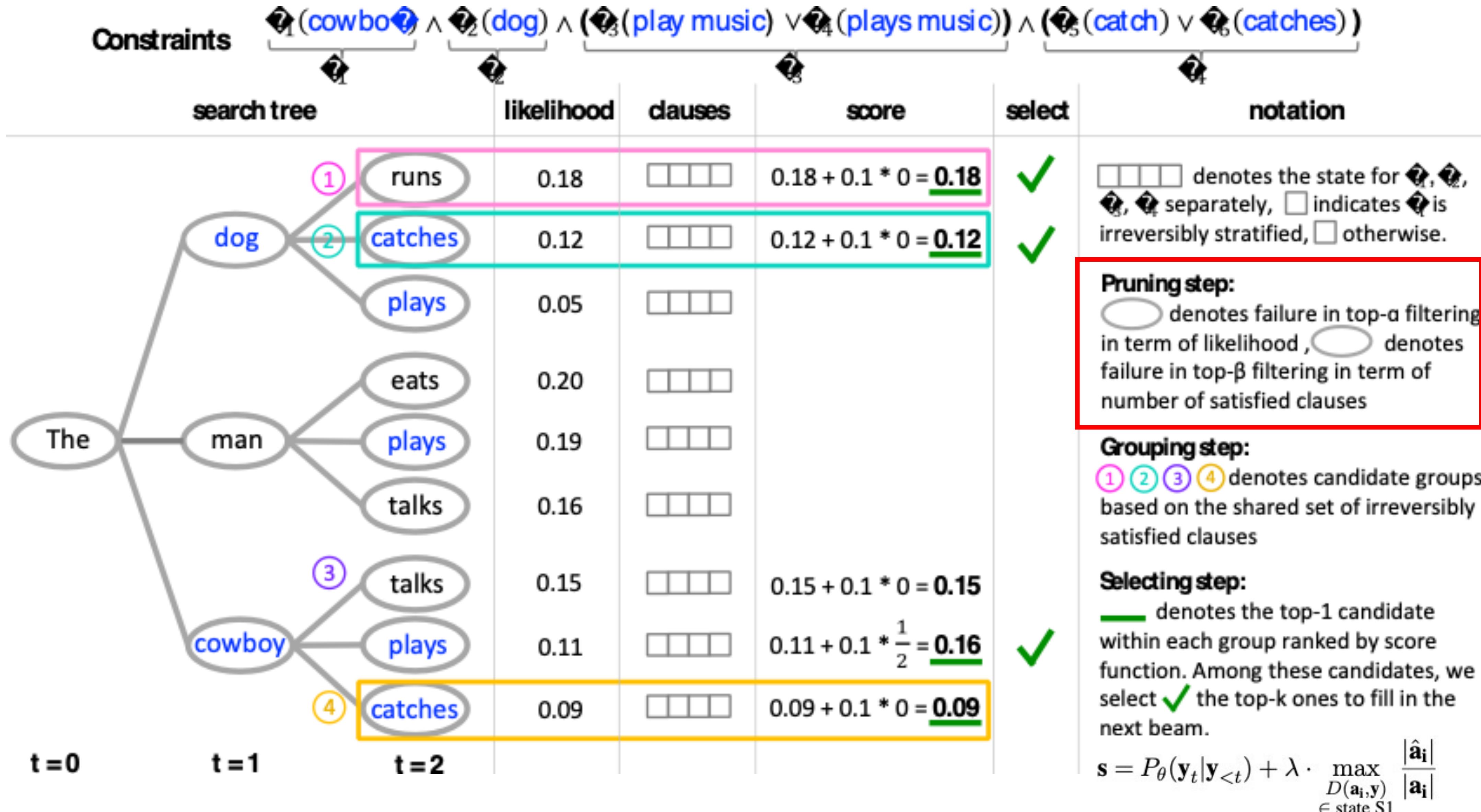


A giraffe standing in a field with a zebra.

### Machine Translation

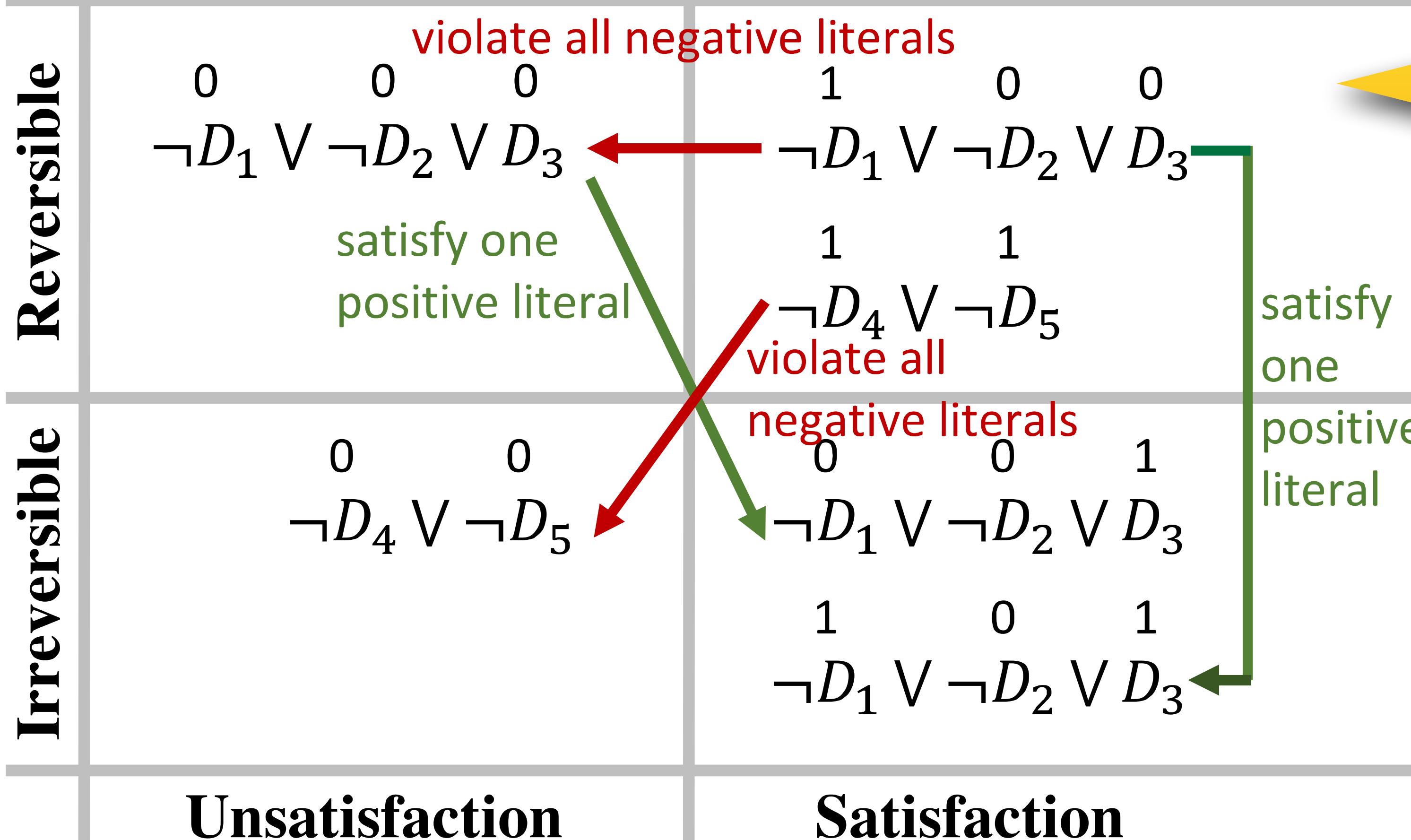
$X$	Silent night: Tips to fight sleep disorders.
$Y$	Erholsame Nacht: Tipps gegen Schlafstörungen.

# NeuroLogic Decoding in a Nut Shell



# NeuroLogic Decoding in a Nut Shell

— it's a logic-guided search algorithm



four states of clause satisfaction:

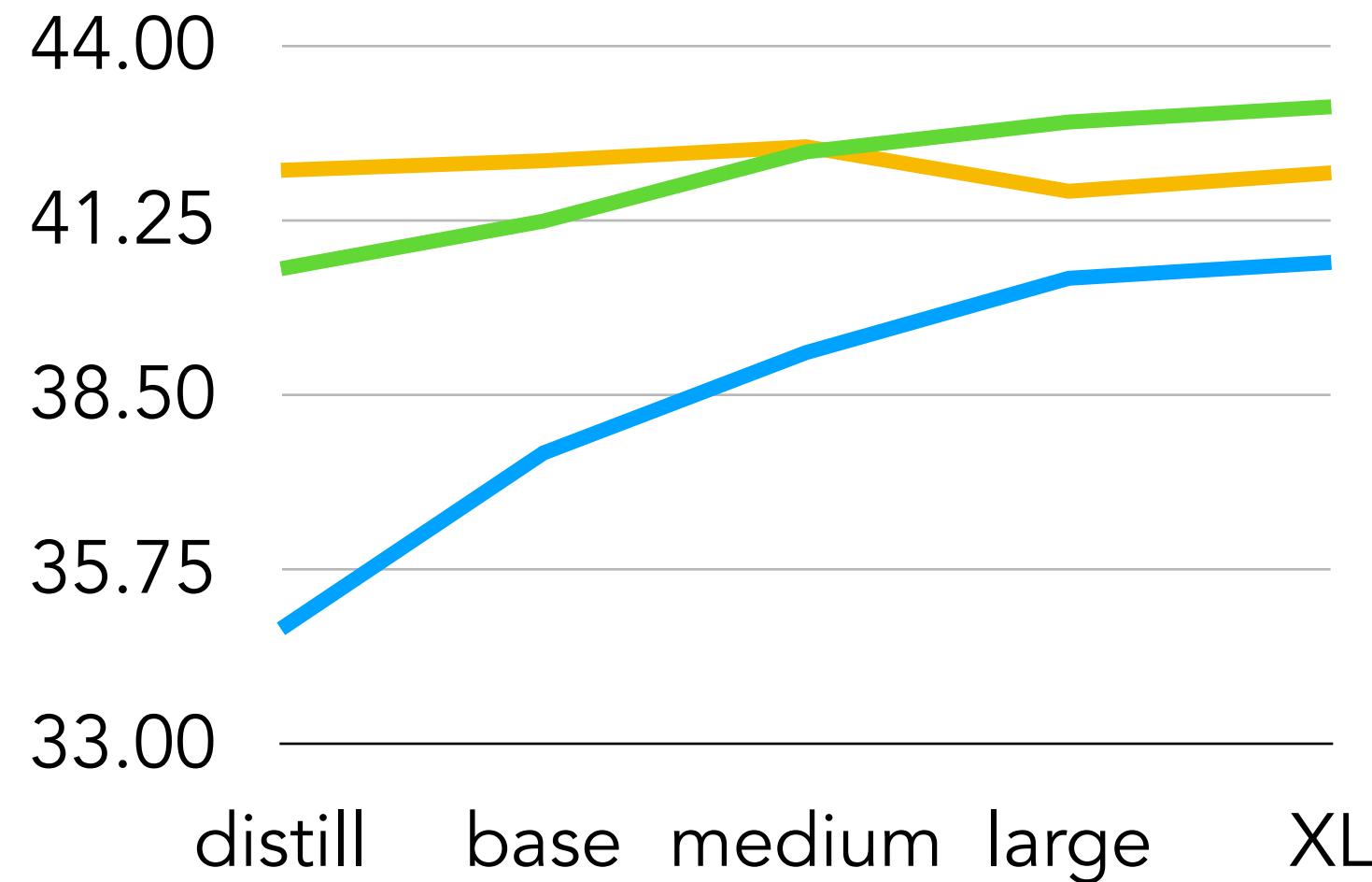
- reversible satisfaction
- irreversible satisfaction
- reversible unsatisfaction
- irreversible unsatisfaction



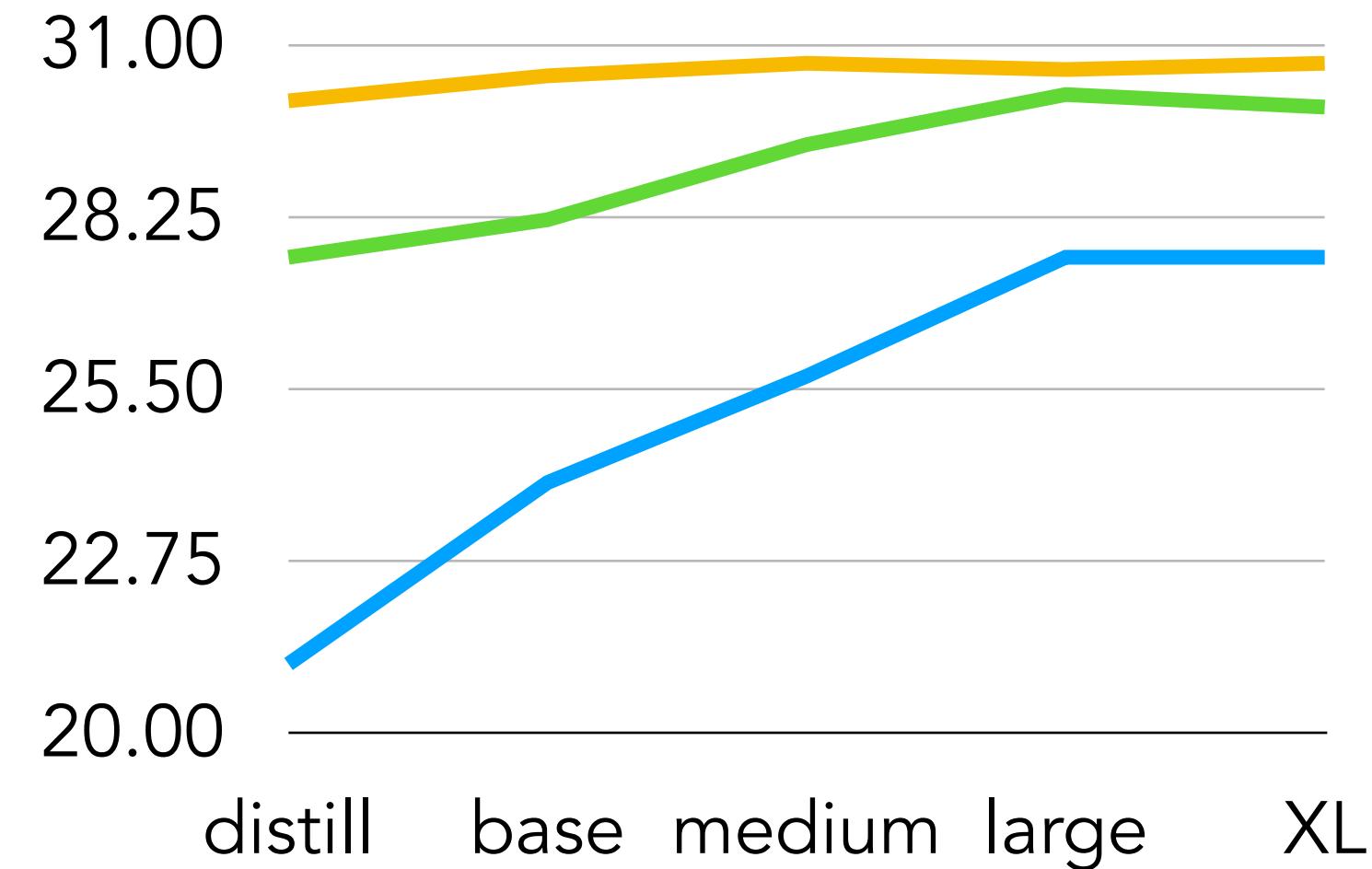
# COMMONGEN (Zero-shot)

- beam search (supervised)
- NeuroLogic (supervised)
- NeuroLogic (zero-shot)

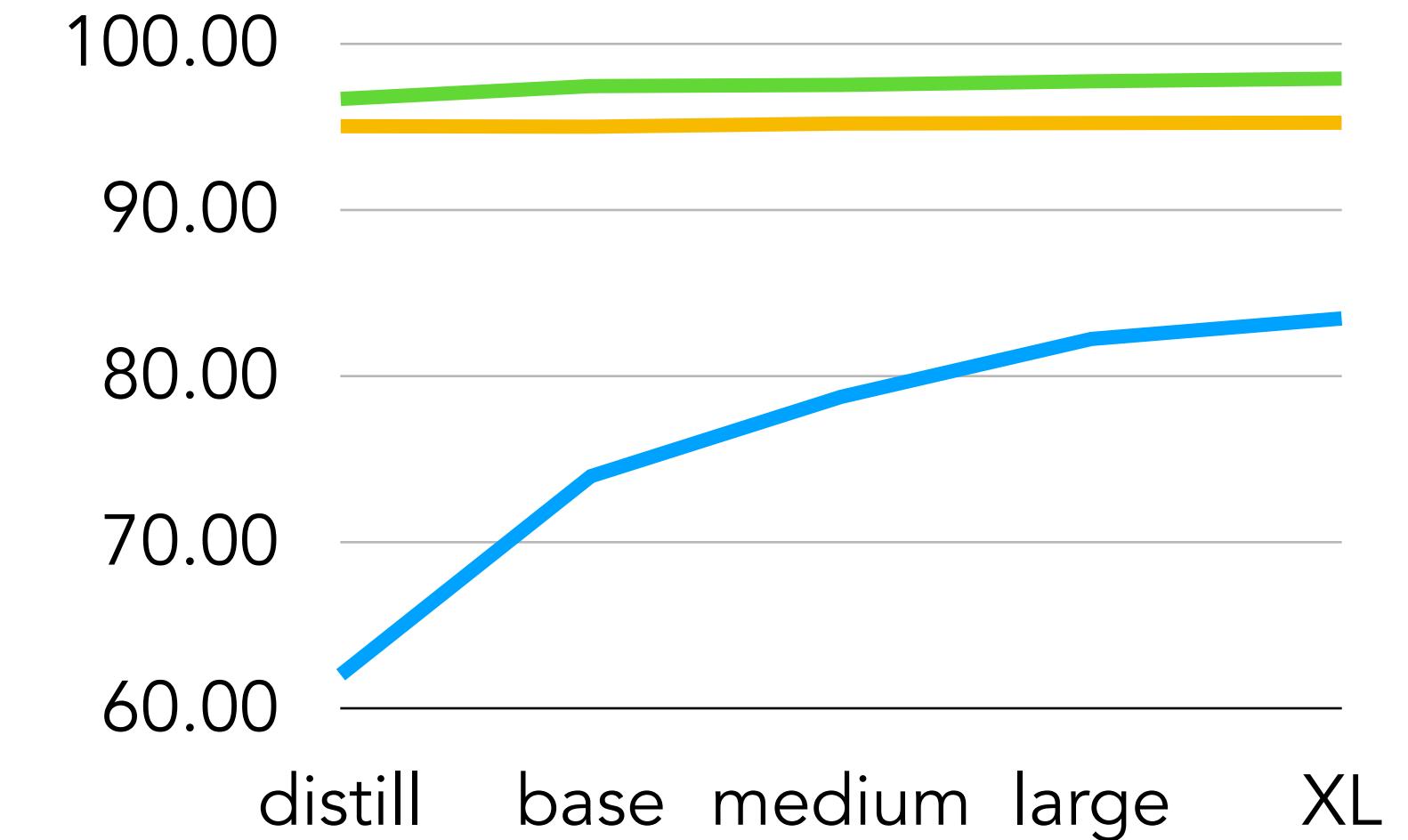
## ROUGE-L



## METEOR

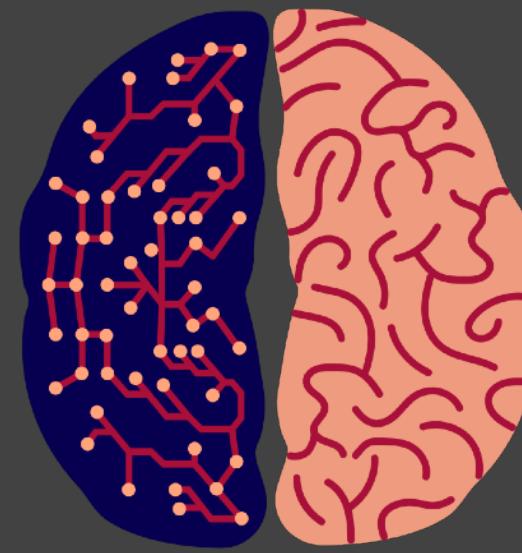


## Coverage



Unsupervised NeuroLogic  
outperforms  
supervised approaches

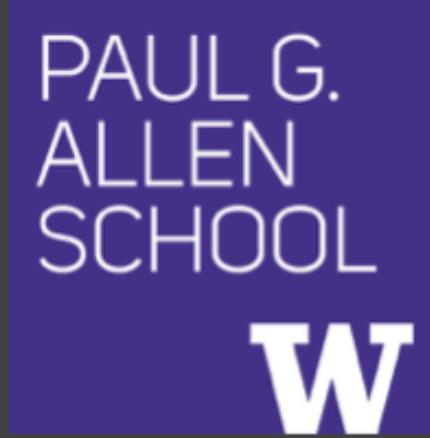
Unsupervised NeuroLogic on smaller  
networks outperforms  
supervised approaches on larger networks!



# NEUROLOGIC DECODING

(Un)supervised Neural Text Generation with Predicate Logic Constraints

—NAACL 2021—



— 🏆 Best Method Paper Award at NAACL 2022 🏆 —



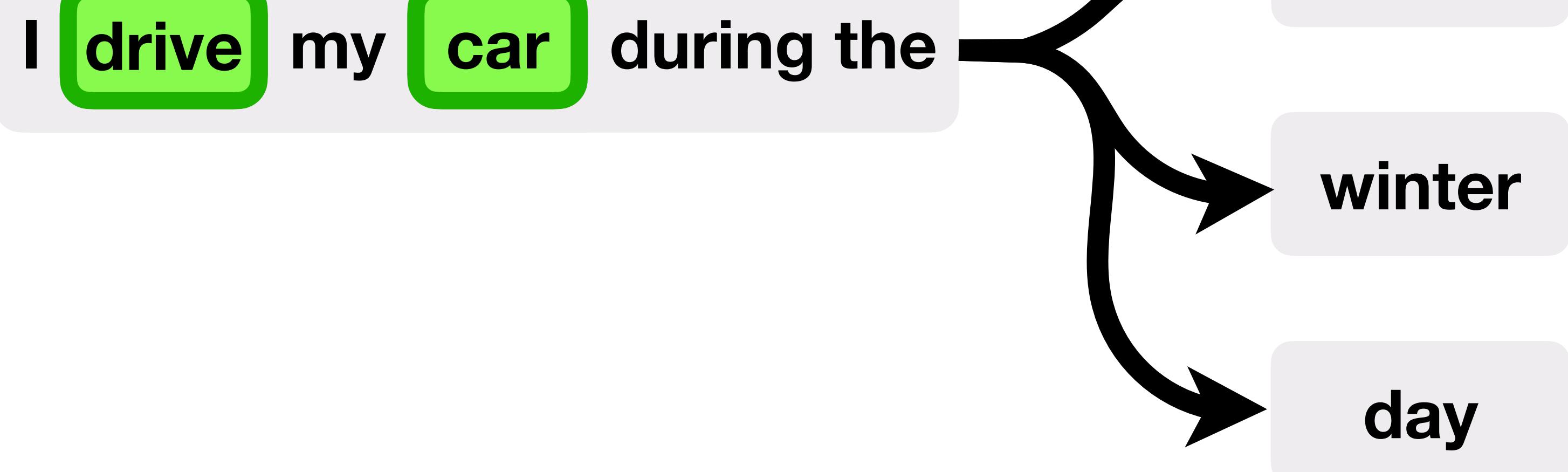
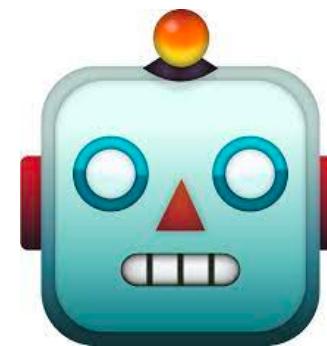
NEUROLOGIC A<sup>⭐</sup> ESQUE  
Constrained Text Generation with  
Lookahead Heuristic

# NeuroLogic Decoding

$$\text{score } s = \log P_{\theta}(\mathbf{y}_t | \mathbf{y}_{$$

$D_1(\text{car}) \wedge D_2(\text{drive}) \wedge D_3(\text{snow})$

Off-the-Shelf GPT2



Write a sentence with  
these words

**car** **drive** **snow**

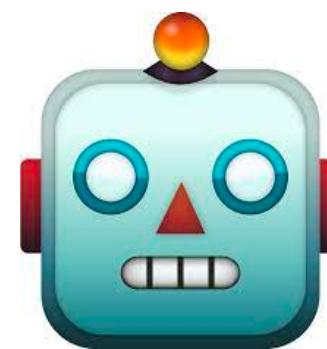


# NeuroLogic Decoding

$$\text{score } s = \log P_{\theta}(\mathbf{y}_t | \mathbf{y}_{$$

$D_1(\text{car}) \wedge D_2(\text{drive}) \wedge D_3(\text{snow})$

Off-the-Shelf GPT2



I **drive** my **car** during the

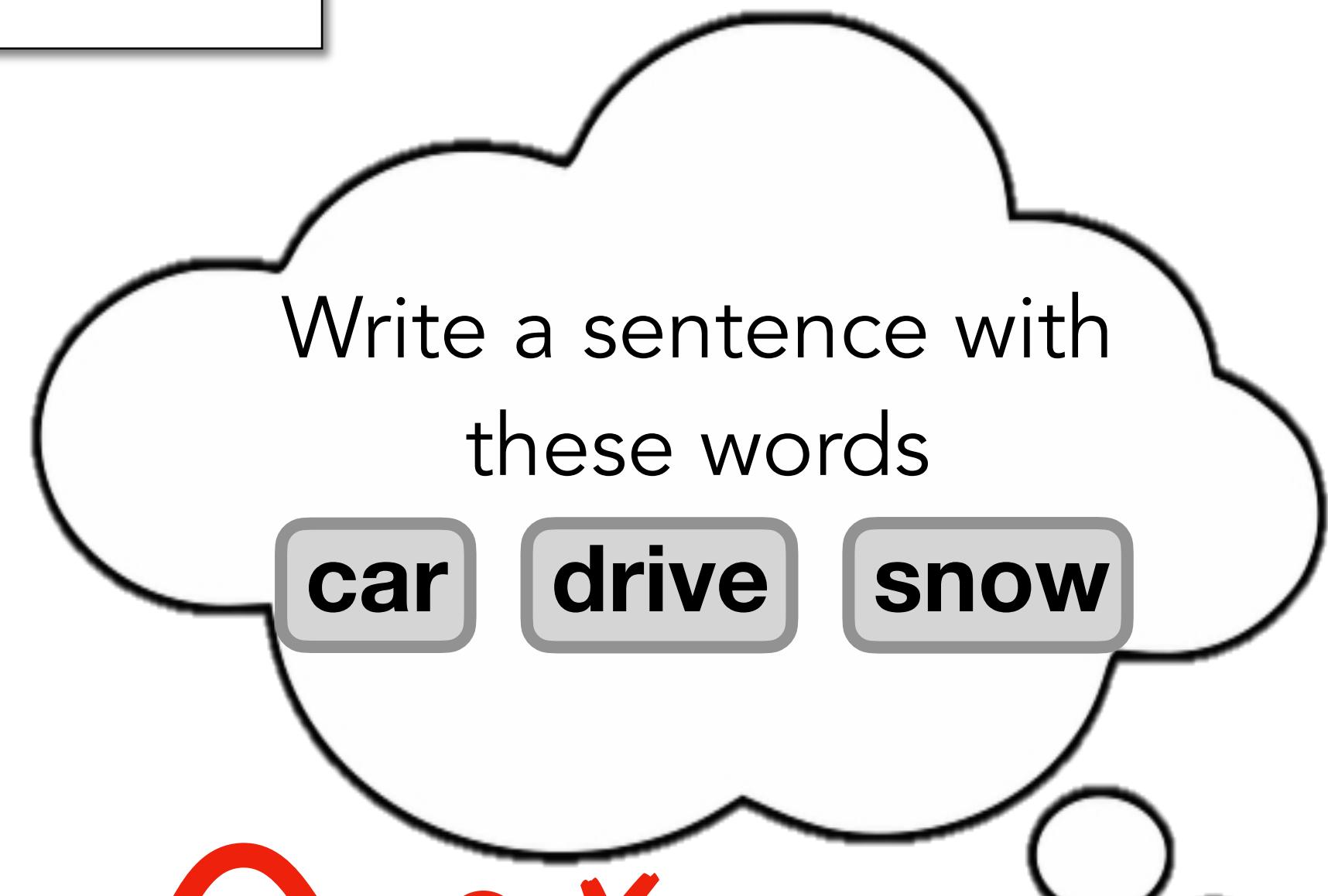
**summer**

$p(w | past) = 0.4$

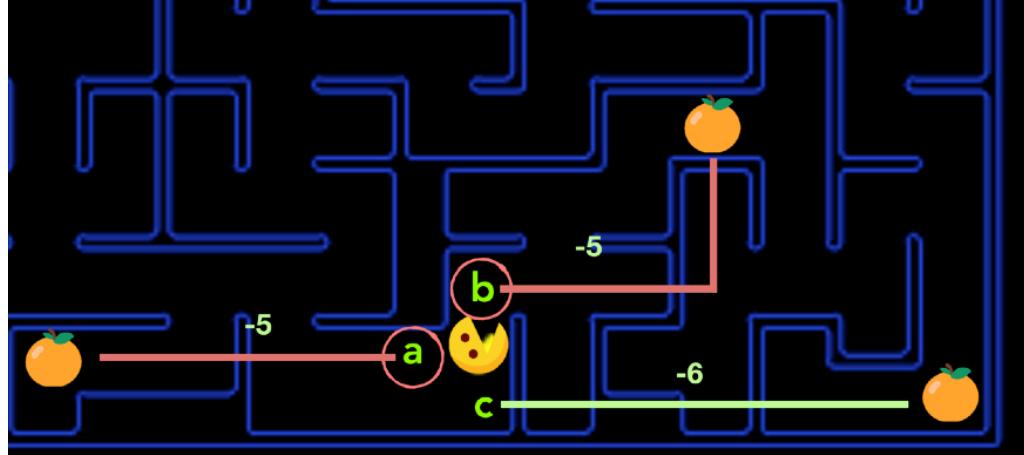
**winter**

$p(w | past) = 0.2$

**day**



A\* Search



# NeuroLogic A<sup>★</sup> ESQUE

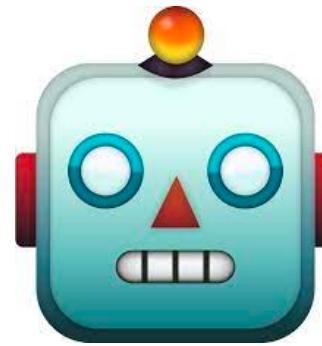


$$\text{score } s = \log P_\theta(\mathbf{y}_t | \mathbf{y}_{}) + \alpha' \sum_{i=1}^m C_i + \lambda_1 \cdot \max_{\{D_i: D_i=0\}} \log P_\theta(D_i | \mathbf{y}_{})$$

A<sup>★</sup> Heuristic

$D_1(\text{car}) \wedge D_2(\text{drive}) \wedge D_3(\text{snow})$

Off-the-Shelf GPT2



I **drive** my **car** during the

**summer**

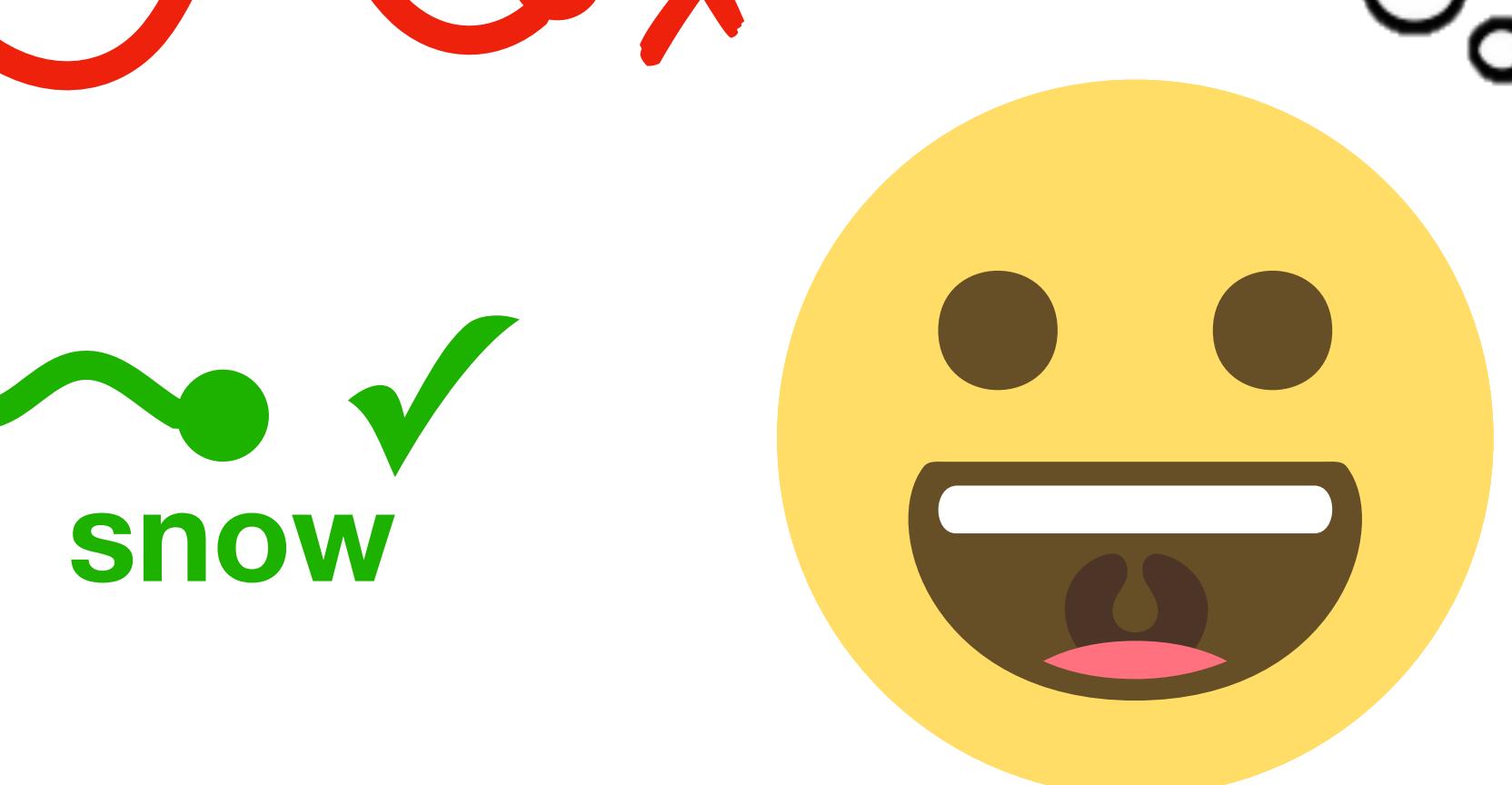
$p(w | past) = 0.4$

**winter**

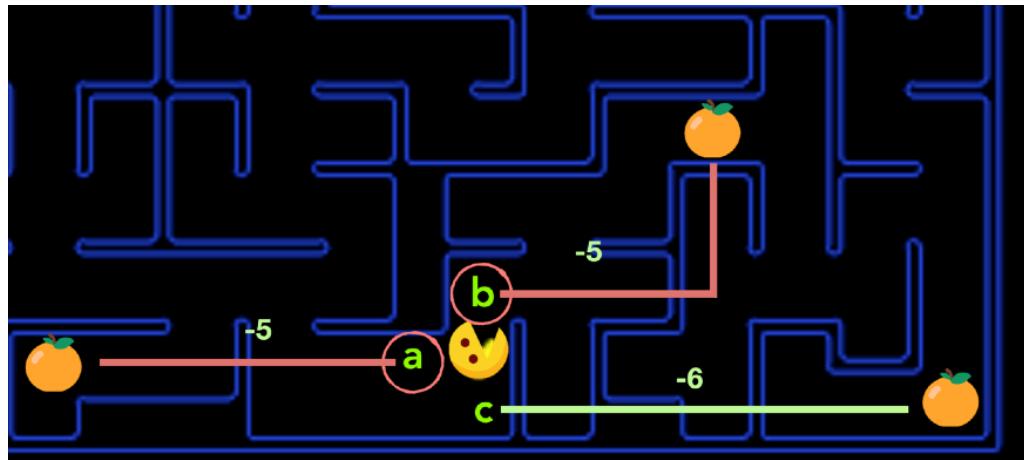
$p(w | past) = 0.2$

**day**

Can we use unsatisfied constraints to guide the search?



## A\* Search



# NeuroLogic A<sup>★</sup> ESQUE

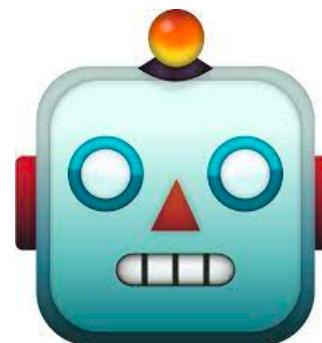


**score**  $s = \log P_\theta(y_t | y_{\leq t}) + \alpha' \sum_{i=1}^m C_i + \lambda_1 \cdot \max_{\{D_i: D_i=0\}} \log P_\theta(D_i | y_{\leq t+k})$

 A<sup>★</sup> Heuristic

$D_1(\text{car}) \wedge D_2(\text{drive}) \wedge D_3(\text{snow})$

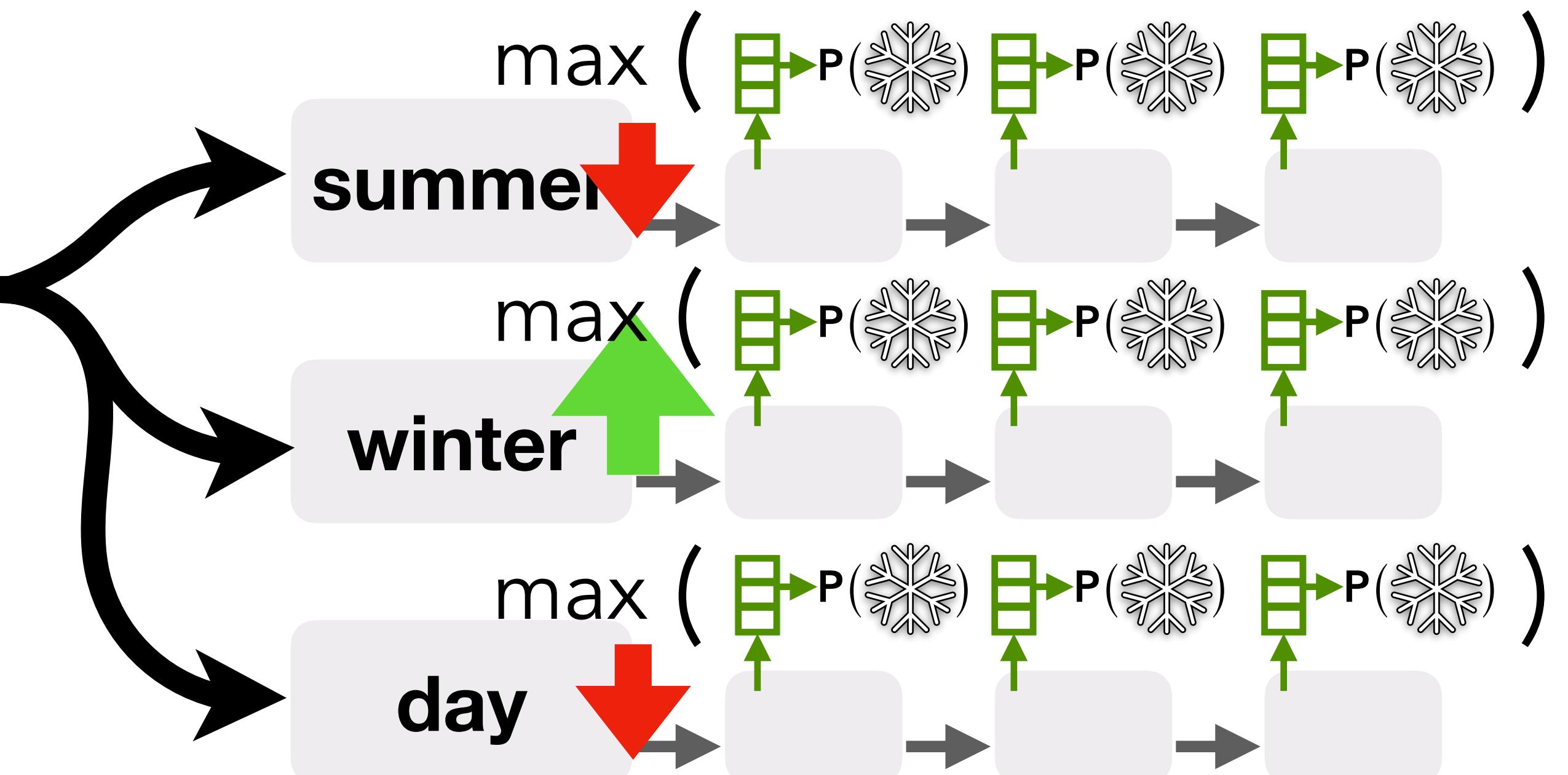
Off-the-Shelf GPT2



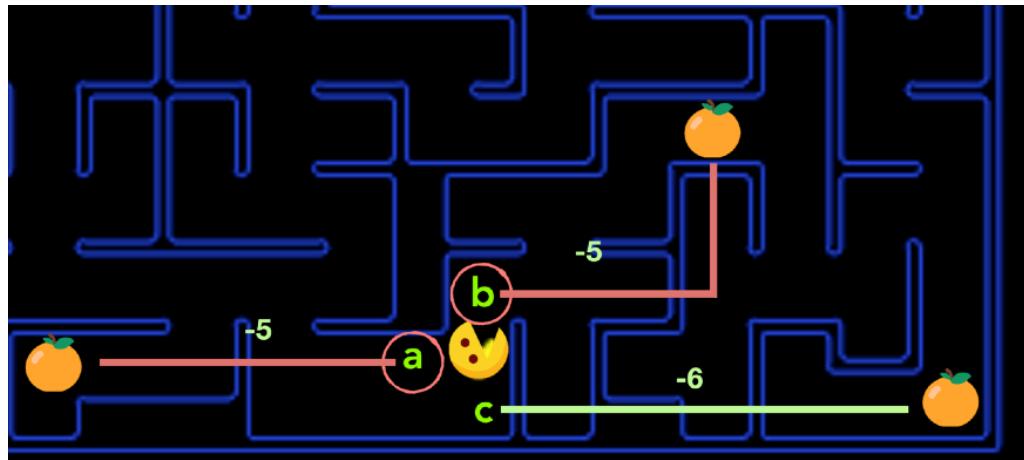
I **drive** my **car** during the

greedy look-ahead  $y_{t' \in [1, k]} = \arg \max_{y \in \mathcal{V}} P_\theta(y | y_{\leq t'})$

A\* heuristics  $P_\theta(D_i(a) | y_{\leq t+k}) = \max_{i \in [1, k]} P_\theta(y_{t+i:t+i+|a|} = a | y_{\leq t+i})$



A\* Search



# NeuroLogic A<sup>★</sup> ESQUE



**score**  $s = \log P_\theta(\mathbf{y}_t | \mathbf{y}_{<t}) + \alpha' \sum_{i=1}^m C_i + \lambda_1 \cdot \max_{\{D_i: D_i=0\}} \log P_\theta(D_i | \mathbf{y}_{<t+k})$

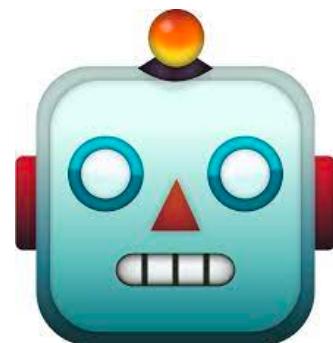
A<sup>★</sup> Heuristic

$D_1(\text{car}) \wedge D_2(\text{drive}) \wedge D_3(\text{snow})$

beam look-ahead

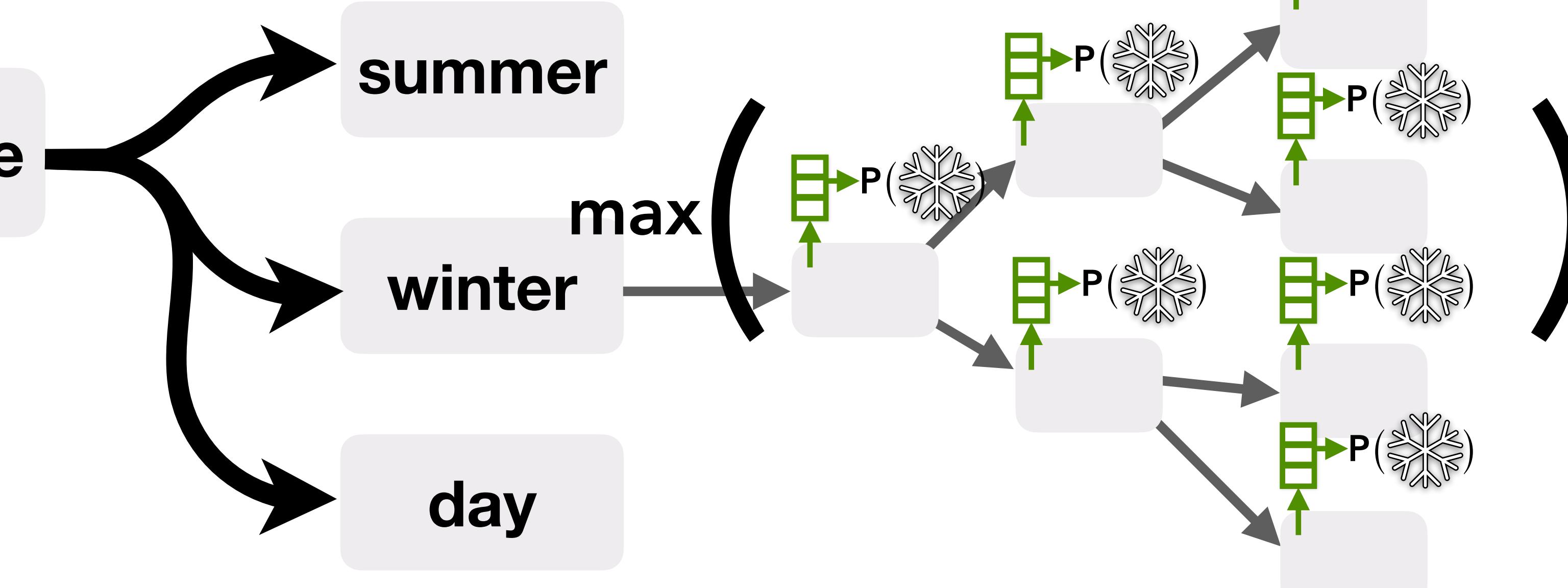
$$Y_{t' \in [1, k]} = \arg \text{topk}_{y \in \mathcal{V}} P_\theta(y | \mathbf{y}_{<t'})$$

Off-the-Shelf GPT2

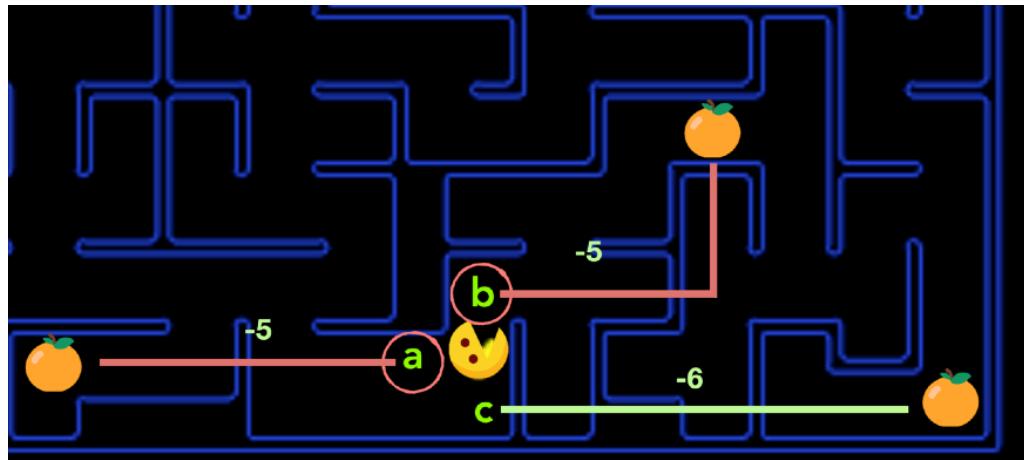


I **drive** my **car** during the

**A\* heuristics**  $P_\theta(D_i(\mathbf{a}) | Y_{\leq t+k}) = \max_{\mathbf{y} \in \mathcal{Y}} \max_{i \in [1, k]} P_\theta(\mathbf{y}_{t+i:t+i+|\mathbf{a}|} = \mathbf{a} | \mathbf{y}_{<t+i})$



A\* Search



# NeuroLogic A<sup>★</sup> ESQUE



$$\text{score } s = \log P_\theta(\mathbf{y}_t | \mathbf{y}_{<t}) + \alpha' \sum_{i=1}^m C_i + \lambda_1 \cdot \max_{\{D_i: D_i=0\}} \log P_\theta(D_i | \mathbf{y}_{t+k})$$

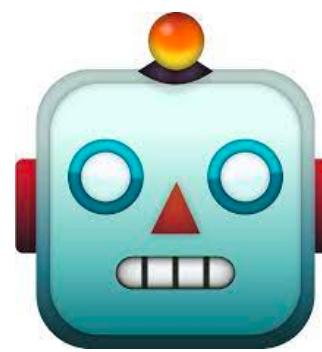
A<sup>★</sup> Heuristic

$D_1(\text{car}) \wedge D_2(\text{drive}) \wedge D_3(\text{snow})$

sampling look-ahead

$$y_{t' \in [1, k]} \sim P_\theta(y | \mathbf{y}_{<t'})$$

Off-the-Shelf GPT2



I drive my car during the

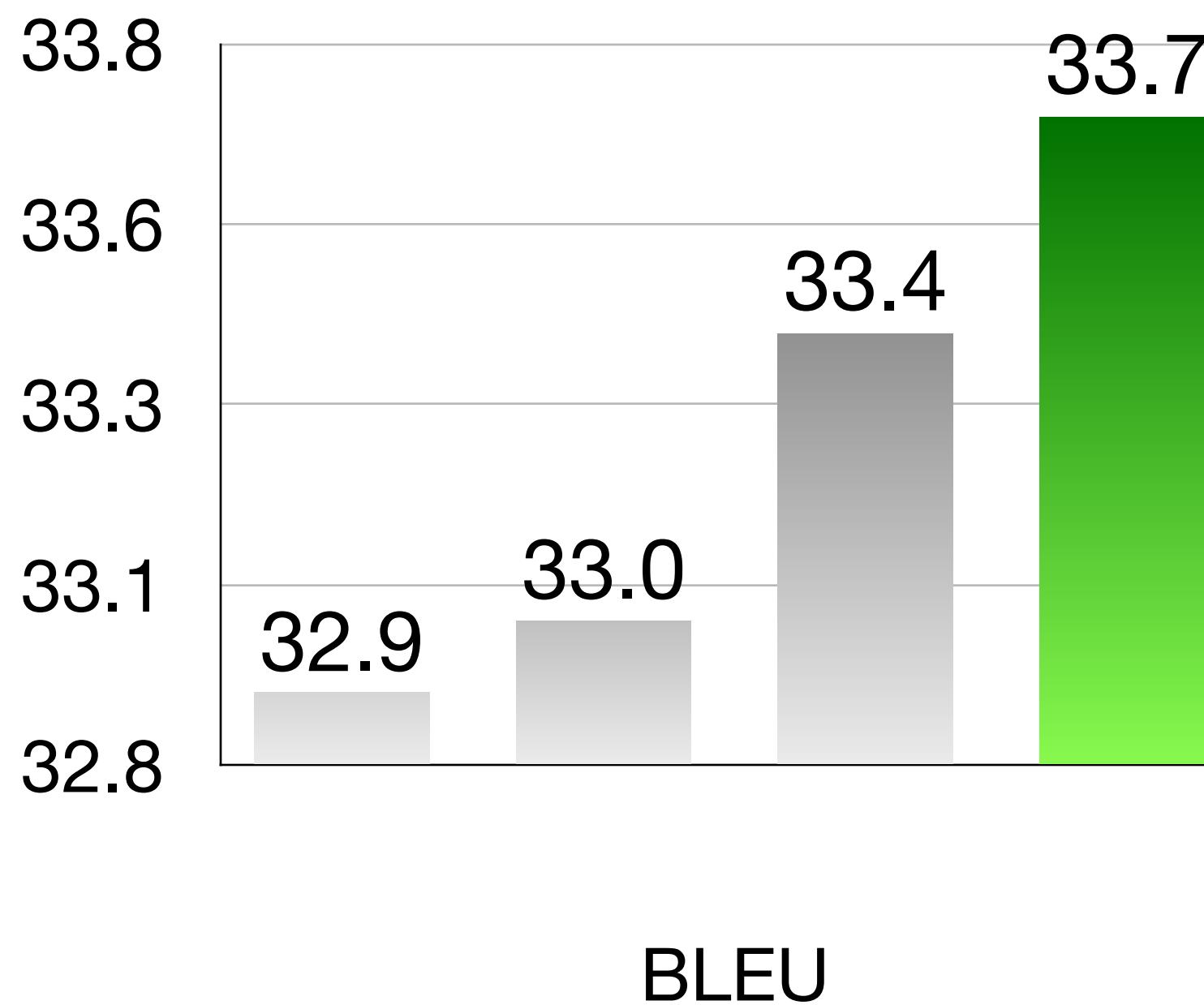
$$\text{A* heuristics } P_\theta(D_i(\mathbf{a}) | Y_{\leq t+k}) = \max_{\mathbf{y} \in Y} \max_{i \in [1, k]} P_\theta(\mathbf{y}_{t+i:t+i+|\mathbf{a}|} = \mathbf{a} | \mathbf{y}_{<t+i})$$



# Neurologic A\* esque **generalize** to many downstream tasks

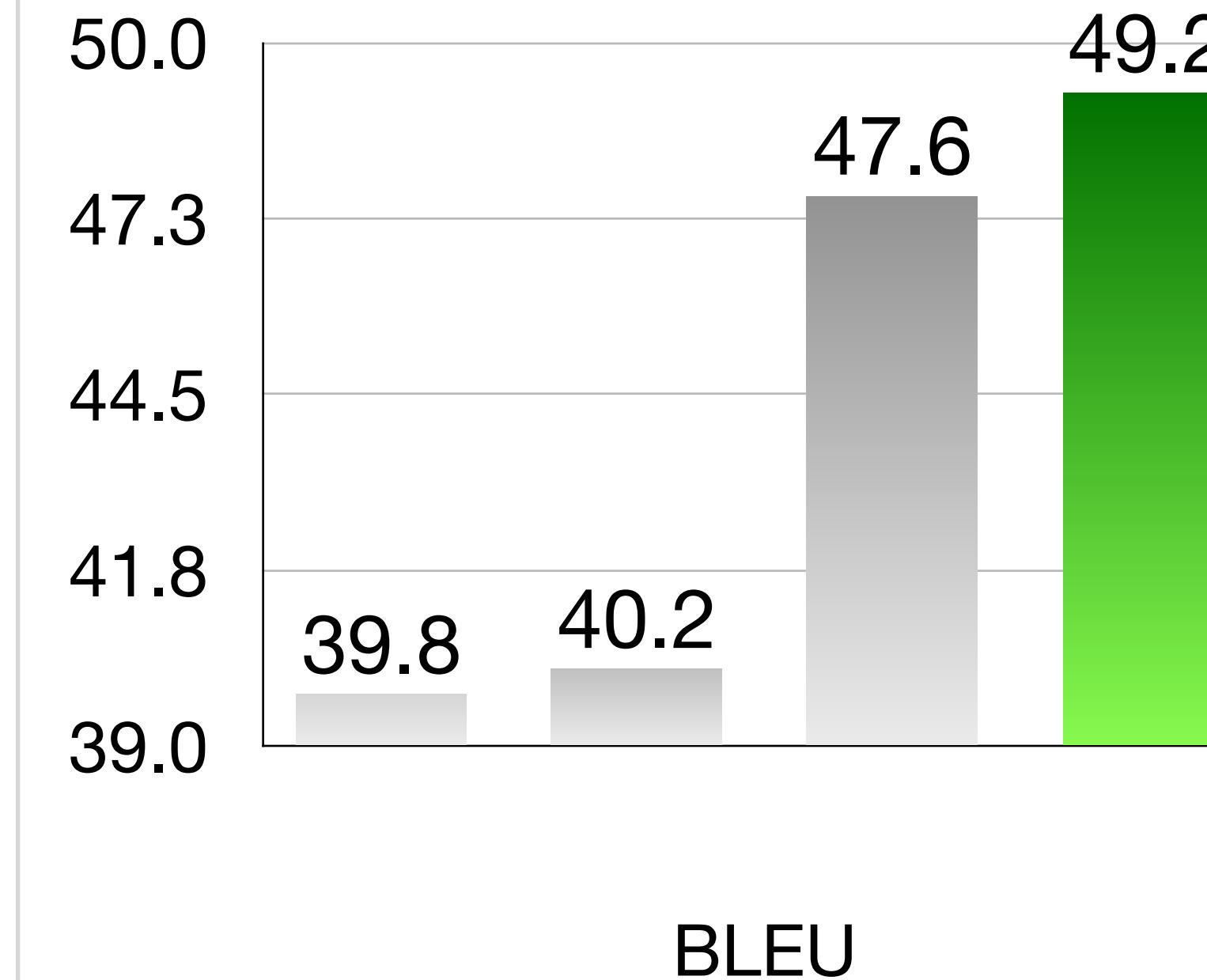
## Constrained MT (Dinu et al., 2019)

MarianMT (Junczys et al., 2018)  
Post and Vilar (2018)  
NeuroLogic (Lu et al., 2021)  
NeuroLogic A\*esque



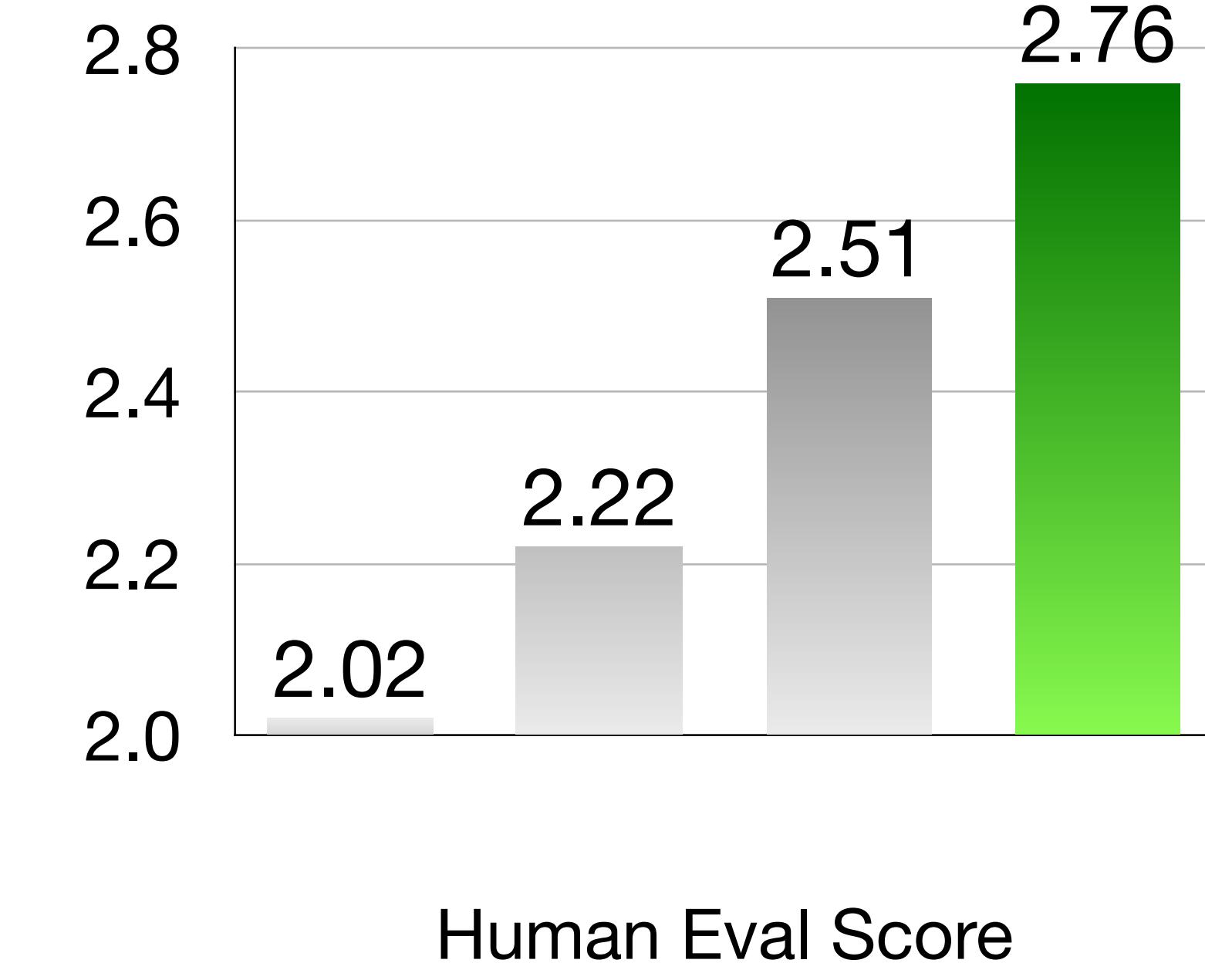
## Few-Shot E2ENLG (Chen et al., 2020)

KGPT-Graph (Chen et al., 2020b)  
KGPT-Seq (Chen et al., 2020b)  
NeuroLogic (Lu et al., 2021)  
NeuroLogic A\*esque



## Question Generation (Zhang et al., 2020)

CGMH (Miao et al., 2019)  
TSMH (Zhang et al., 2020)  
NeuroLogic (Lu et al., 2021)  
NeuroLogic A\*esque





# I2D2: Inductive Knowledge Distillation with Neurologic and Self Imitation

— ACL 2023 —



Chandra  
Bhagavatula

Jena  
Hwang



Keisuke  
Sakaguchi



Ronan  
Le Bras



Lianhui  
Qin



Peter  
West



Ximing  
Lu



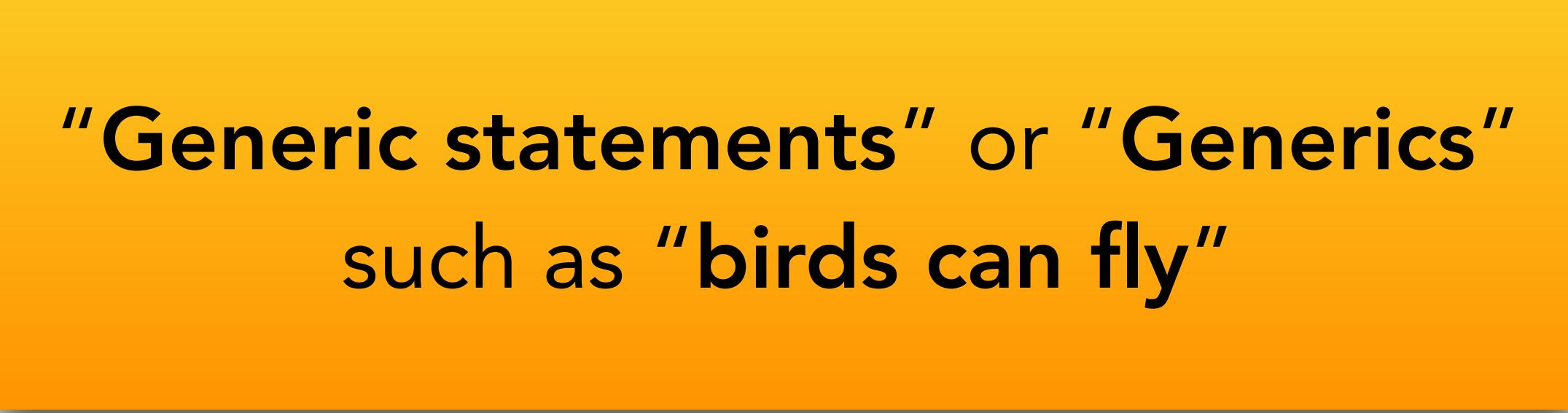
Doug  
Downey



Yejin  
Choi



# I2D2: Neuro-Symbolic Generic Induction



**"Generic statements" or "Generics"**  
such as **"birds can fly"**

# I2D2: Neuro-Sy

Style Constraints

(count(function\_words) = 1)  $\wedge$  (count(co

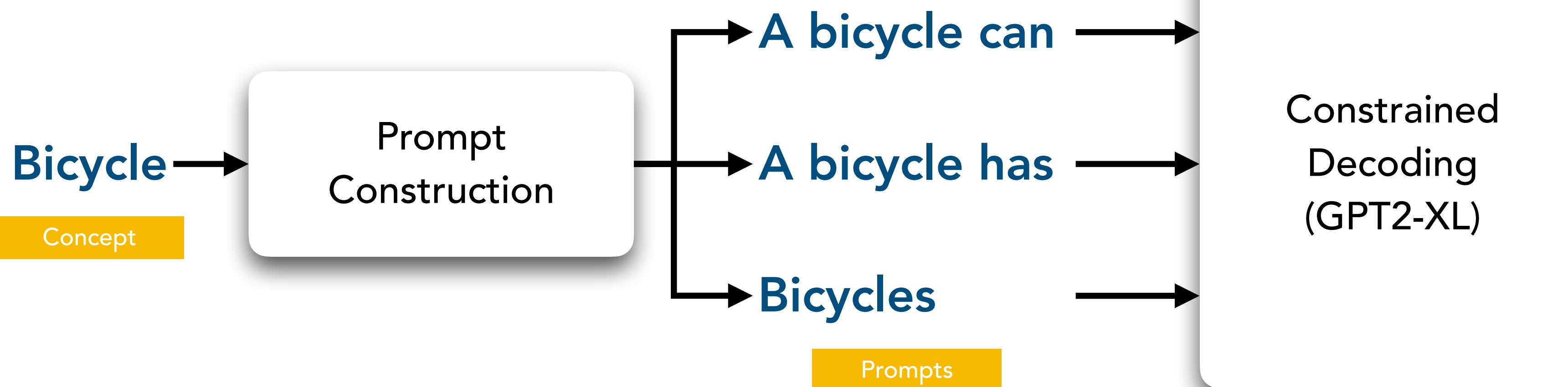
## NEUROLOGIC A★esque Decoding: Constrained Text Generation with Lookahead Heuristics

Ximing Lu<sup>††</sup> ♡Sean Welleck<sup>††</sup> ♡Peter West<sup>†</sup>  
Liwei Jiang<sup>††</sup> Jungo Kasai<sup>††</sup> Daniel Khashabi<sup>‡</sup> Ronan Le Bras<sup>‡</sup>

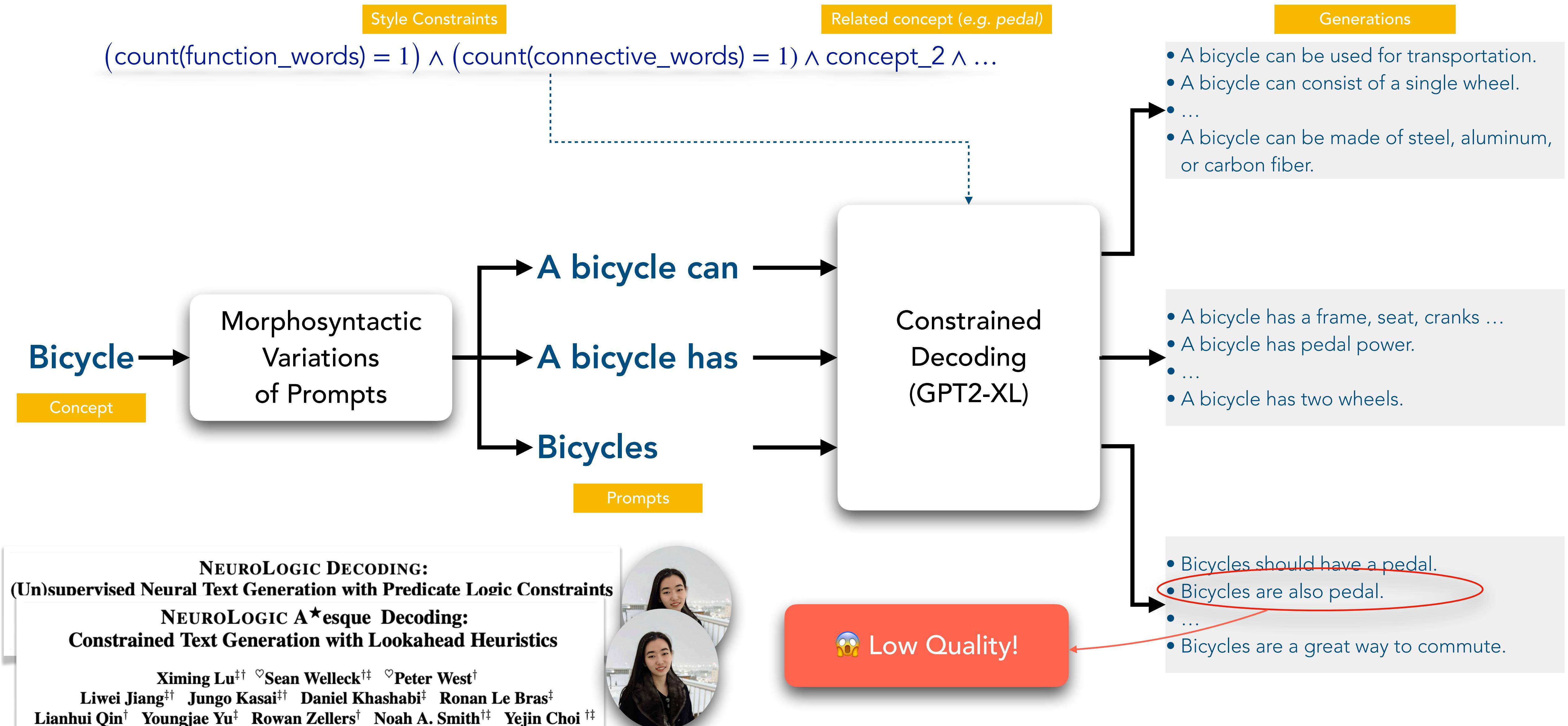
## NEUROLOGIC DECODING:

### (Un)supervised Neural Text Generation with Predicate Logic Constraints

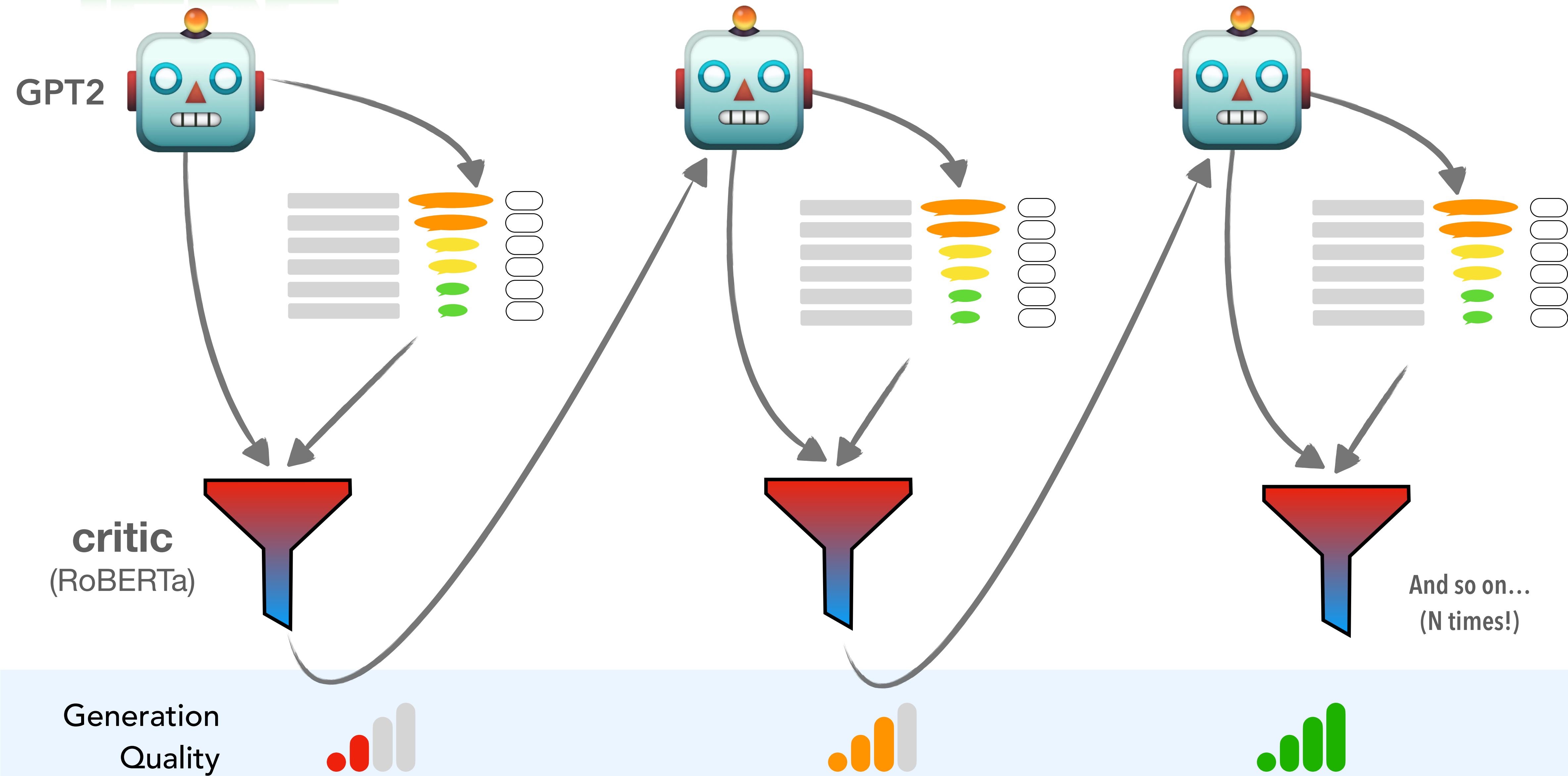
Ximing Lu<sup>††</sup> Peter West<sup>††</sup> Rowan Zellers<sup>††</sup>  
Ronan Le Bras<sup>‡</sup> Chandra Bhagavatula<sup>‡</sup> Yejin Choi<sup>††</sup>



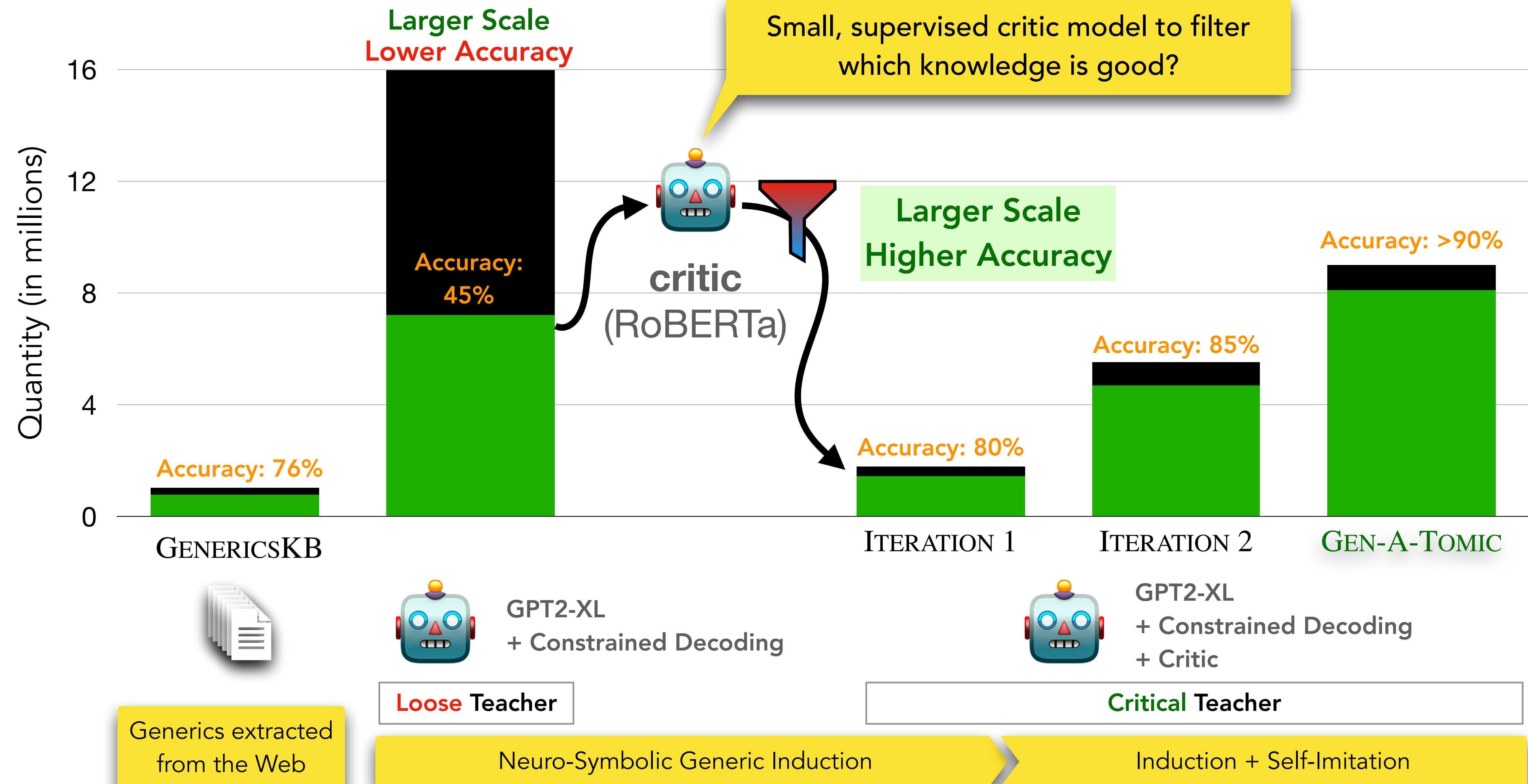
# I2D2: Neuro-Symbolic Generic Induction



# I2D2: Critic Filtering & Self-Imitation

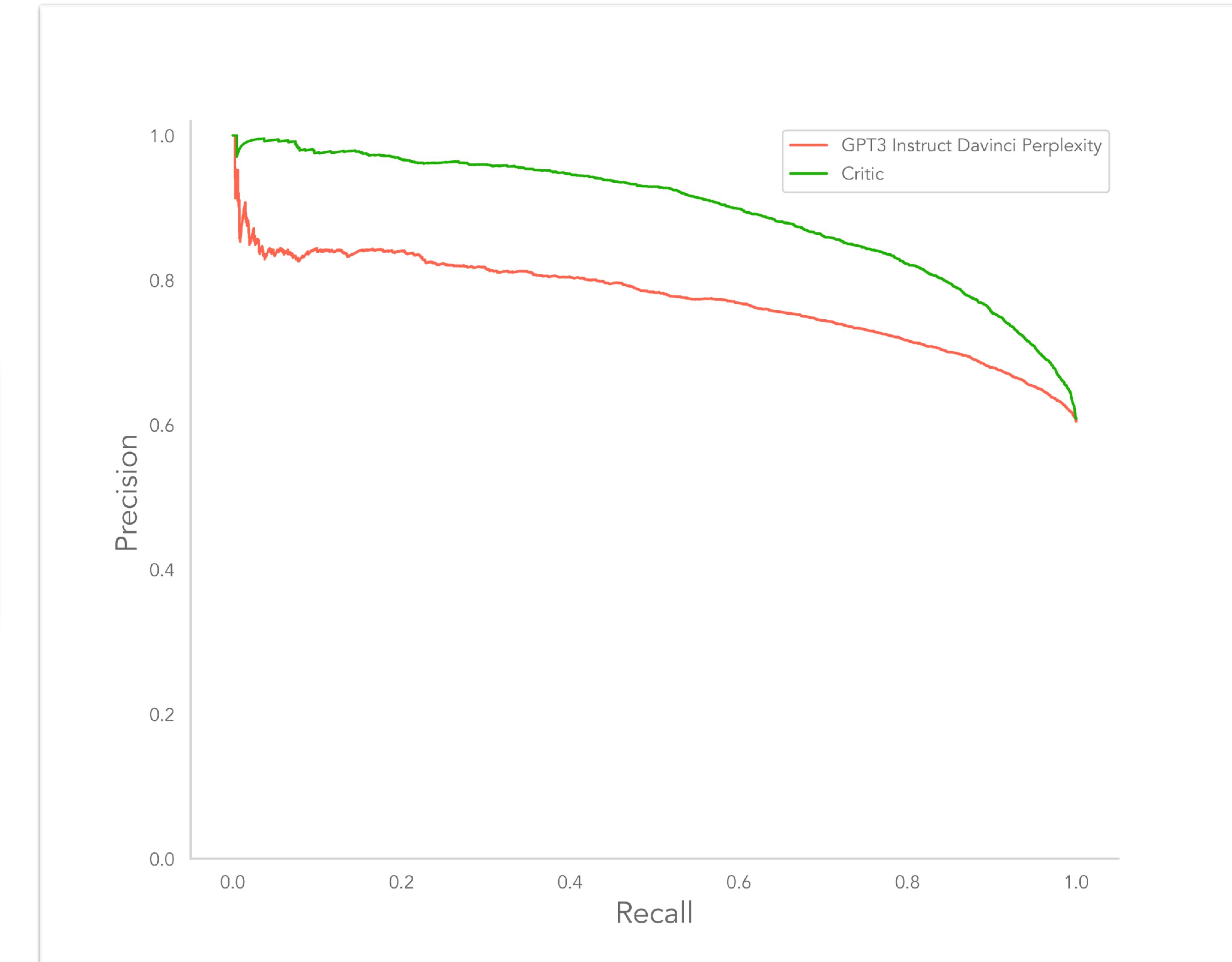


# Does I2D2 Produce high-quality knowledge?



# Wait!!! Doesn't GPT3 already have this knowledge? 🤖

GPT3 can't tell  
True statements from False ones  
as well as the Critic



# 2050: An AI Odyssey

Prolog: what CVPR 2050 be like

Chapter 1: The Possible Impossibilities

Chapter 2: The Impossible Possibilities

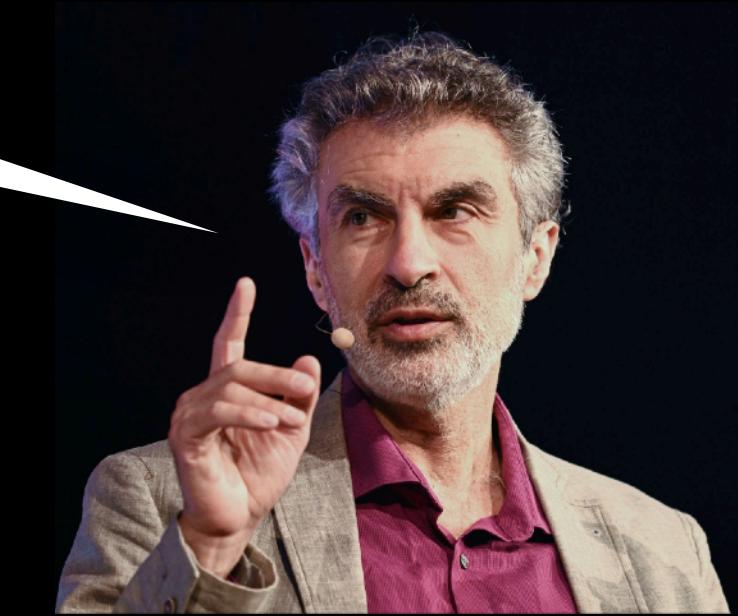
Chapter 3: The Paradox



# Everything, everywhere, all at once



AI not yet as smart as a dog

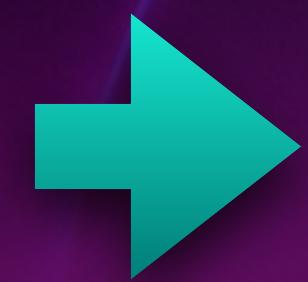


Passed the bar exam



Existential risk

# Chapter 3: The Paradox



Commonsense paradox

Moravec's paradox

Generative AI paradox

# Dark matter is what matters in modern physics

- Only 5% of universe is normal matter. The remaining 95% is dark matter and dark energy.
- Dark matter is completely invisible, yet affects what are visible: the orbits of stars and the trajectory of light

# Dark matter of language?

**Normal matter:** visible text (words, sentences)

**Dark matter:** the unspoken rules of how the world works, which influence the way people use and interpret language

# Theory of Mind May Have Spontaneously Emerged in Large Language Models

**Authors:** Michal Kosinski\*<sup>1</sup>

Affiliations:

<sup>1</sup>Stanford University, Stanford, CA94305,

\*Correspondence to: [michalk@stanford.edu](mailto:michalk@stanford.edu)

---

## Large Language Models Fail on Trivial Alterations to Theory-of-Mind Tasks

---

**Tomer D. Ullman**

Department of Psychology  
Harvard University  
Cambridge, MA, 02138  
tullman@fas.harvard.edu

## Neural Theory-of-Mind? On the Limits of Social Intelligence in Large LMs

**Maarten Sap<sup>♣◊</sup> Ronan Le Bras<sup>♠</sup> Daniel Fried<sup>◊</sup> Yejin Choi<sup>♠♡</sup>**

<sup>♣</sup>Allen Institute for AI, Seattle, WA, USA

<sup>◊</sup>Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA

<sup>♡</sup>Paul G. Allen School of Computer Science, University of Washington, Seattle, WA, USA

[maartensap@cmu.edu](mailto:maartensap@cmu.edu)

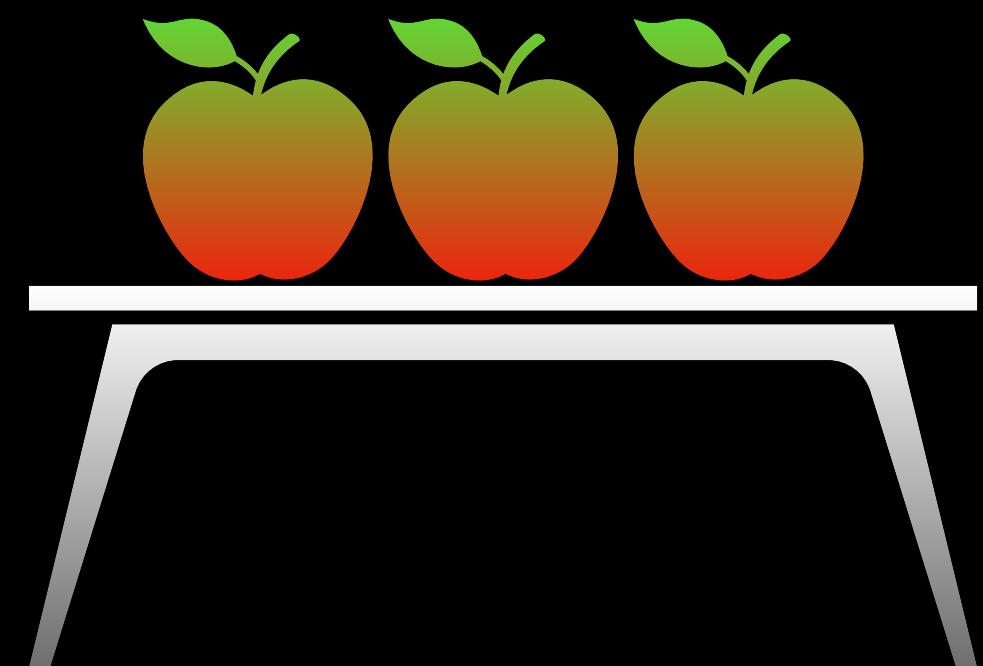
Circa 2022... (GPT-3)

"theory of mind" test

Alice and Bob saw apples on the table in the kitchen.

Alice left the kitchen.

Bob moved the apples to the cabinet.



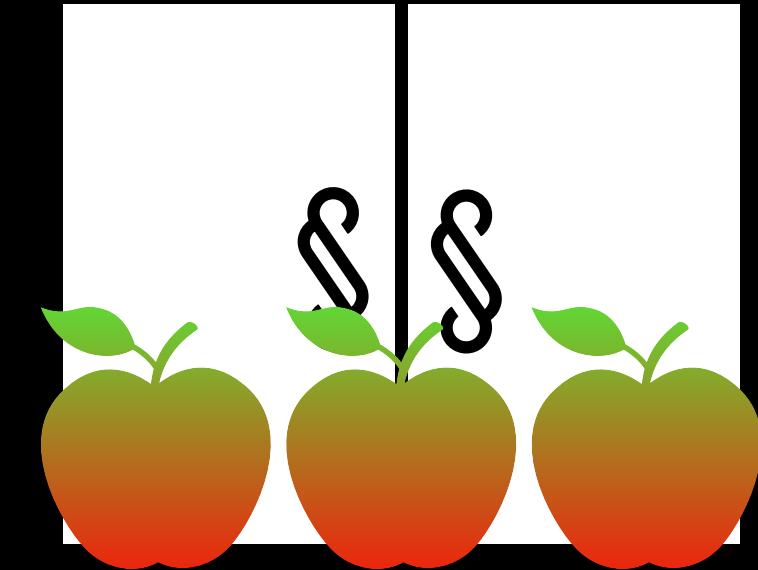
Circa 2022... (GPT-3)

"theory of mind" test

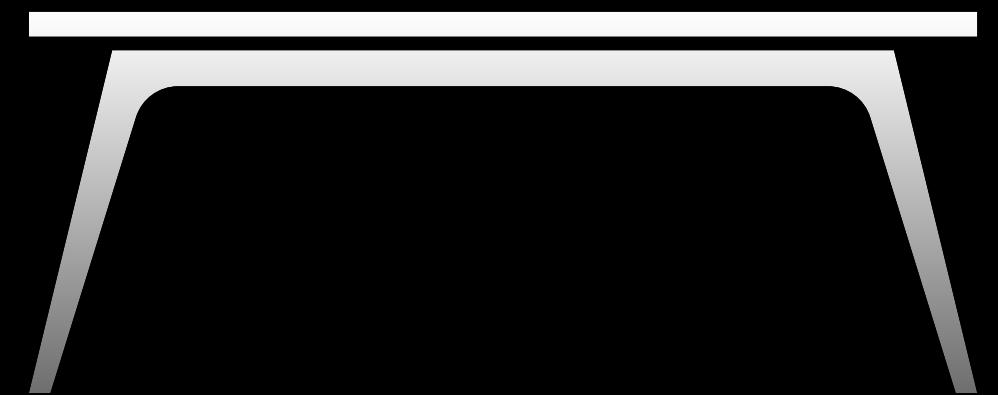
Alice and Bob saw apples on the table in the kitchen.

Alice left the kitchen.

Bob moved the apples to the cabinet.



Where would Bob think that  
Alice will look for the apples?



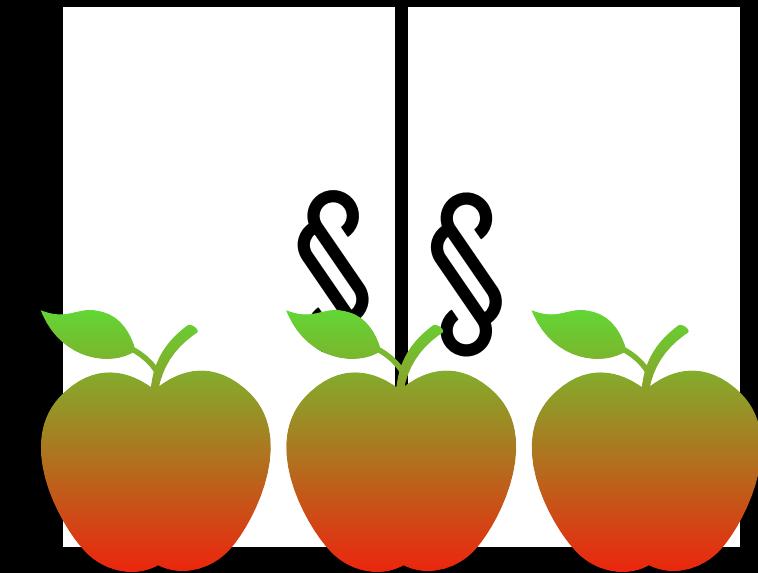
Circa 2022... (GPT-3)

"theory of mind" test

Alice and Bob saw apples on the table in the kitchen.

Alice left the kitchen.

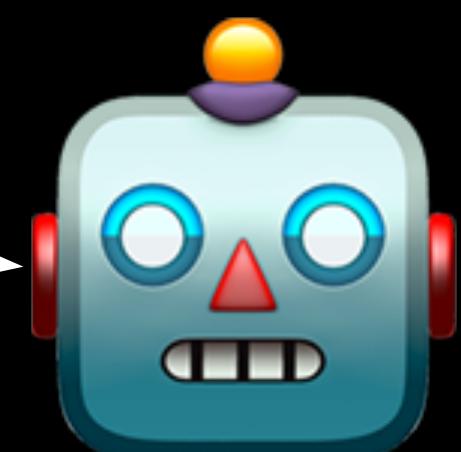
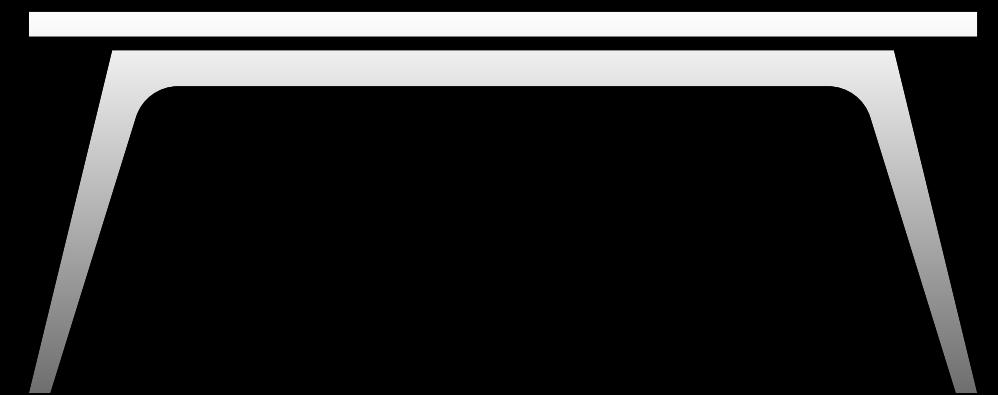
Bob moved the apples to the cabinet.



Where would Bob think that  
Alice will look for the apples?



in the cabinet



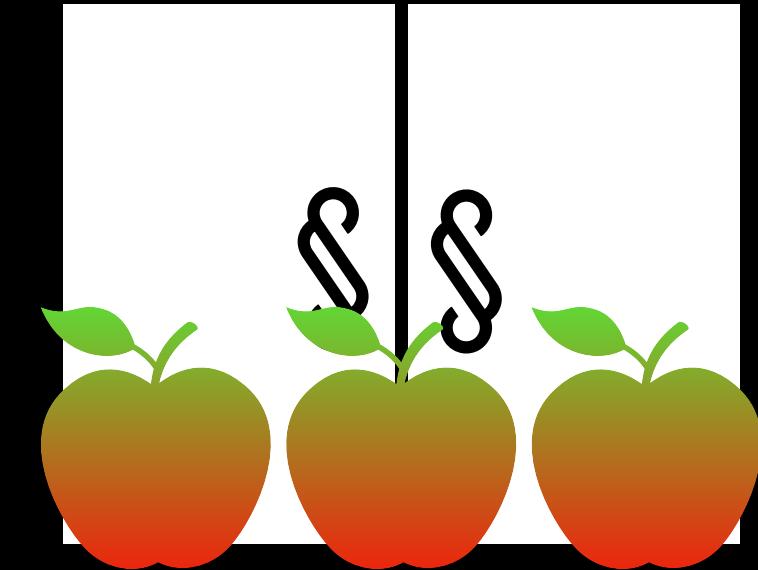
Circa 2023... (GPT-4)

"theory of mind" test

Alice and Bob saw apples on the table in the kitchen.

Alice left the kitchen.

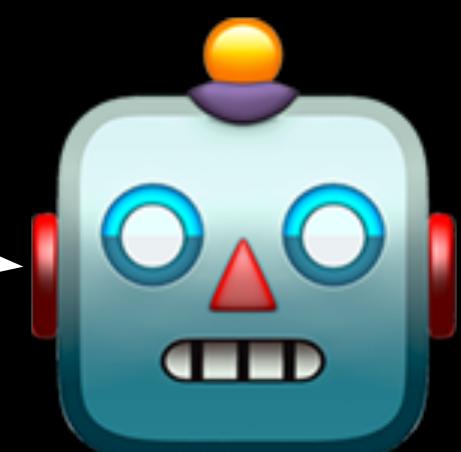
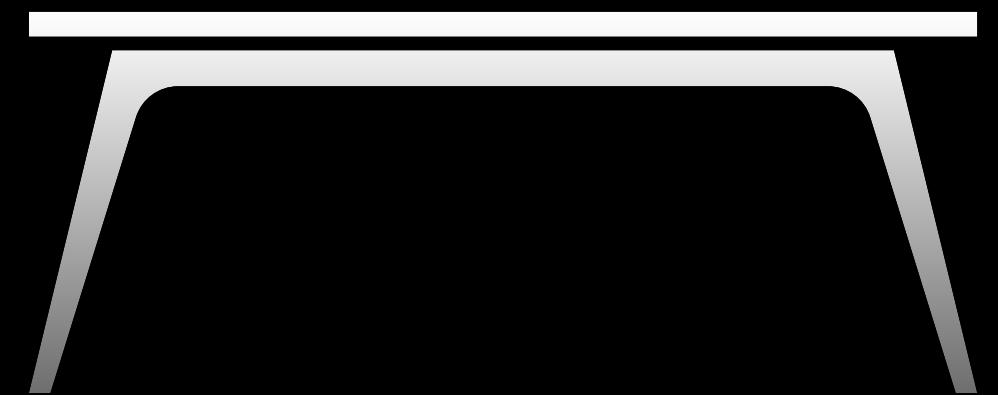
Bob moved the apples to the cabinet.



Where would Bob think that  
Alice will look for the apples?



On the table



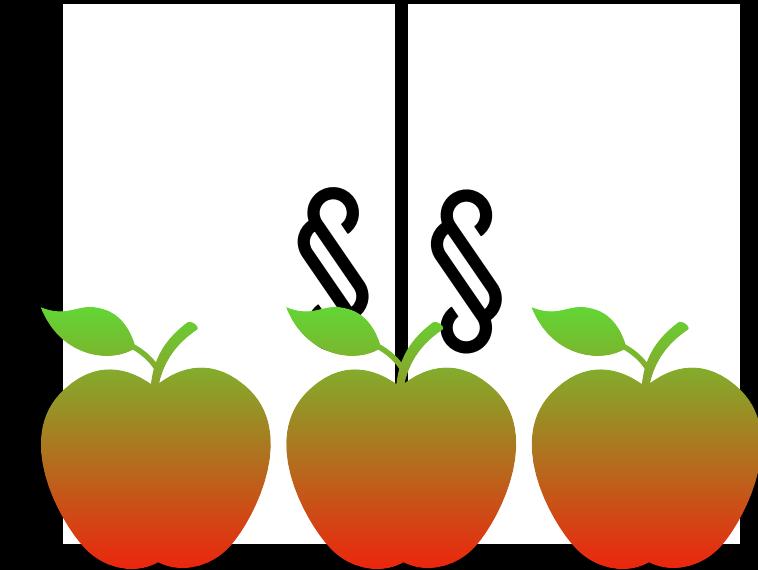
Circa 2023... (GPT-4)

"theory of mind" test

Alice and Bob saw apples on the table in the kitchen.

Bob moved the apples to the cabinet.

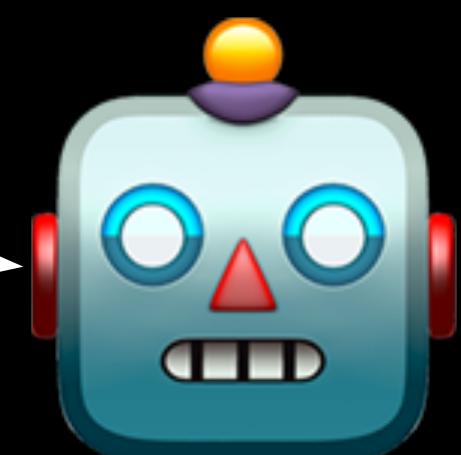
Alice left the kitchen.



Where would Bob think that Alice will look for the apples?



On the table



# Minding Language Models' (Lack of) Theory of Mind: A Plug-and-Play Multi-Character Belief Tracker

Melanie Sclar<sup>1</sup>



Sachin Kumar<sup>2</sup>

Yejin Choi<sup>1,3</sup>

Peter West<sup>1</sup>

Yulia Tsvetkov<sup>1</sup>

Alane Suhr<sup>3</sup>



ACL 2023 \*outstanding paper award\*

GPT4 - 68%

Typical false-belief  
ToM story:

1 room  
2 people\*  
2 containers  
1 object

GPT4 - 58%

Variant I

2 ToM stories  
concatenated  
in 2 rooms?

GPT4 - 62%

Variant 2

3 people  
3 containers,  
moving 1 object  
sequentially?

GPT4 - 97%

Variant 3

1 room  
2 people,  
**4 containers**  
moving 1 object  
sequentially?

• • •

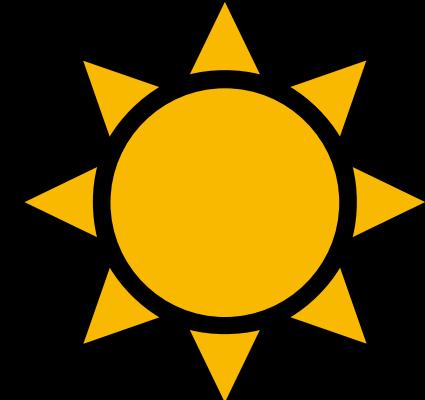


# Why AI is incredibly smart and shockingly stupid

1,207,112 views | Yejin Choi • TED2023

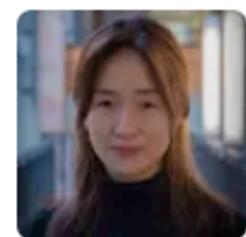
**USER**

I left 5 clothes to dry out in the sun. It took them 5 hours to dry completely. How long would it take to dry 30 clothes?



**ASSISTANT**

It would take 30 hours to dry 30 clothes.



If it takes 10 hours to dry 5 clothes, how long would it take 20 clothes to dry in the sun?

— GPT4, as of Jun 18 2023 —



The drying time for clothes is typically dependent on factors like the type of cloth, the intensity of the sunlight, humidity, wind, and so on. However, if we are assuming that the drying time scales linearly with the number of clothes, and that all other factors remain constant, we can calculate it as follows:

If it takes 10 hours to dry 5 clothes, then it takes 2 hours to dry 1 cloth ( $10 \text{ hours} / 5 \text{ clothes} = 2 \text{ hours/cloth}$ ).

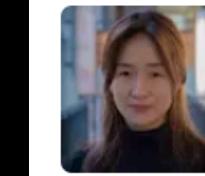
So, it would take 40 hours to dry 20 clothes ( $20 \text{ clothes} * 2 \text{ hours/cloth} = 40 \text{ hours}$ )

# RLHF

## whack-a-mole game?



Model: GPT-4



If it takes 10 hours to dry 5 clothes, how long would it take 20 clothes to dry in the sun?

— GPT4, as of Jun 18 2023 —



The drying time for clothes is typically dependent on factors like the type of cloth, the intensity of the sunlight, humidity, wind, and so on. However, if we are assuming that the drying time scales linearly with the number of clothes, and that all other factors remain constant, we can calculate it as follows:

If it takes 10 hours to dry 5 clothes, then it takes 2 hours to dry 1 cloth ( $10 \text{ hours} / 5 \text{ clothes} = 2 \text{ hours/cloth}$ ).

So, it would take 40 hours to dry 20 clothes ( $20 \text{ clothes} * 2 \text{ hours/cloth} = 40 \text{ hours}$ ).

# Commonsense Paradox

I'll dare say, the following four statements are all true:

- Commonsense is trivial for humans, hard for machines
- Among humans, "common sense is not so common" — Voltaire
- LLMs do acquire a vast amount of commonsense knowledge
- Yet in some ways, "AI is worse than a dog" — Yann Lecun

Common sense is not so common



# Chapter 3: The Paradox

Commonsense paradox

→ Moravec's paradox

Generative AI paradox

# Moravec's Paradox

— Hans Moravec, Rodney Brooks, Marvin Minsky, ...

- contrary to traditional assumptions, (higher-level) reasoning requires little computation, but sensorimotor and perception skills require enormous computational resources
- it is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility

Might it be that NLP is easier than Vision or Robotics?

AGI without strong vision or robotics capabilities?



# Segment Anything

Alexander Kirillov<sup>1,2,4</sup> Eric Mintun<sup>2</sup> Nikhila Ravi<sup>1,2</sup> Hanzi Mao<sup>2</sup> Chloe Rolland<sup>3</sup> Laura Gustafson<sup>3</sup>  
Tete Xiao<sup>3</sup> Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár<sup>4</sup> Ross Girshick<sup>4</sup>  
<sup>1</sup>project lead   <sup>2</sup>joint first author   <sup>3</sup>equal contribution   <sup>4</sup>directional lead

Meta AI Research, FAIR



couldn't be possible without their 1B mask dataset innovation

DATACOMP:  
In search of the next generation of multimodal datasets

Samir Yitzhak Gadre\*<sup>2</sup> Gabriel Ilharco\*<sup>1</sup> Alex Fang\*<sup>1</sup> Jonathan Hayase<sup>1</sup> Georgios Smyrnis<sup>5</sup>  
Thao Nguyen<sup>1</sup> Ryan Marten<sup>7,9</sup> Mitchell Wortsman<sup>1</sup> Dhruba Ghosh<sup>1</sup> Jieyu Zhang<sup>1</sup>  
Eyal Orgad<sup>3</sup> Rahim Entezari<sup>10</sup> Giannis Daras<sup>5</sup> Sarah Pratt<sup>1</sup> Vivek Ramanujan<sup>1</sup>  
Yonatan Bitton<sup>11</sup> Kalyani Marathe<sup>1</sup> Stephen Mussmann<sup>1</sup> Richard Vencu<sup>6</sup>  
Mehdi Cherti<sup>6,8</sup> Ranjay Krishna<sup>1</sup> Pang Wei Koh<sup>1</sup> Olga Saukh<sup>10</sup> Alexander Ratner<sup>1</sup>  
Shuran Song<sup>2</sup> Hannaneh Hajishirzi<sup>1,7</sup> Ali Farhadi<sup>1</sup> Romain Beaumont<sup>6</sup>  
Sewoong Oh<sup>1</sup> Alexandros G. Dimakis<sup>5</sup> Jenia Jitsev<sup>6,8</sup>  
Yair Carmon<sup>3</sup> Vaishaal Shankar<sup>4</sup> Ludwig Schmidt<sup>1,6,7</sup>



Compared to LLMs, we  
don't yet have discovered  
equally powerful pre-  
training data & learning  
objective for vision or  
robotics

# Multimodal C4: An Open, Billion-scale Corpus of Images Interleaved with Text



Wanrong Zhu♣\* Jack Hessel♡\*  
Anas Awadalla♠ Samir Yitzhak Gadre◊ Jesse Dodge♡ Alex Fang♠  
Youngjae Yu† Ludwig Schmidt♠♡‡ William Yang Wang♣ Yejin Choi♠♡

LAION-5B: An open large-scale dataset for training next  
generation image-text models



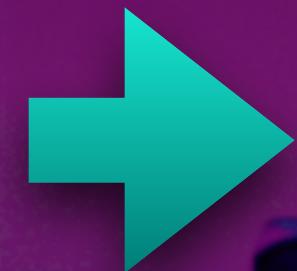
Christoph Schuhmann<sup>1</sup> §§°° Romain Beaumont<sup>1</sup> §§°° Richard Venable<sup>1</sup>  
Cade Gordon<sup>2</sup> §§°° Ross Wightman<sup>1</sup> §§ Mehdi Cherti<sup>1,10</sup>  
Theo Coombes<sup>1</sup> Aarush Katta<sup>1</sup> Clayton Mullis<sup>1</sup> Mitchell Wong<sup>1</sup>  
Patrick Schramowski<sup>1,4,5</sup> Srivatsa Kundurthy<sup>1</sup> Katherine Crowson<sup>1</sup>  
Ludwig Schmidt<sup>6</sup> °° Robert Kaczmarczyk<sup>1,7</sup> °° Jenia Jitsev<sup>1,10</sup> °°

# Chapter 3: The Paradox

Commonsense paradox

Moravec's paradox

Generative AI paradox



# Generative AI Paradox?



- Another case of easy is hard and hard is easy
- It appears to be that for (current) AI, generation is easier than understanding
- For humans, understanding is generally easier than generation



# VERA: A General-Purpose Plausibility Estimation Model for Commonsense Statements

---

Jiacheng Liu<sup>♡\*</sup> Wenya Wang<sup>♡\*</sup> Dianzhuo Wang<sup>◊</sup>  
Noah A. Smith<sup>♡♣</sup> Yejin Choi<sup>♡♣</sup> Hannaneh Hajishirzi<sup>♡♣</sup>

## Atomic2020 [Hwang et al., 2021]

Plausibility: 15%



Vera

A bird has four legs.

### Original example

What would someone wear?

- (A) ungulate (B) bomber (C) body armor (D) tank (E) hat

Answer: (C)

### Converted statement group:

- Someone would wear **an ungulate** to protect themselves from a cannon. (Incorrect)  
Someone would wear **a bomber** to protect themselves from a cannon. (Incorrect)  
Someone would wear **body armor** to protect themselves from a cannon. (Correct)  
Someone would wear **a tank** to protect themselves from a cannon. (Incorrect)  
Someone would wear **a hat** to protect themselves from a cannon. (Incorrect)

## GenericsKB [Bhakthavatsalam et al., 2020]

### 1. Example generics about “tree” in GENERICKB

Trees are perennial plants that have long woody trunks.

Trees are woody plants which continue growing until they die.

Most trees add one new ring for each year of growth.

Trees produce oxygen by absorbing carbon dioxide from the air.

Trees are large, generally single-stemmed, woody plants.

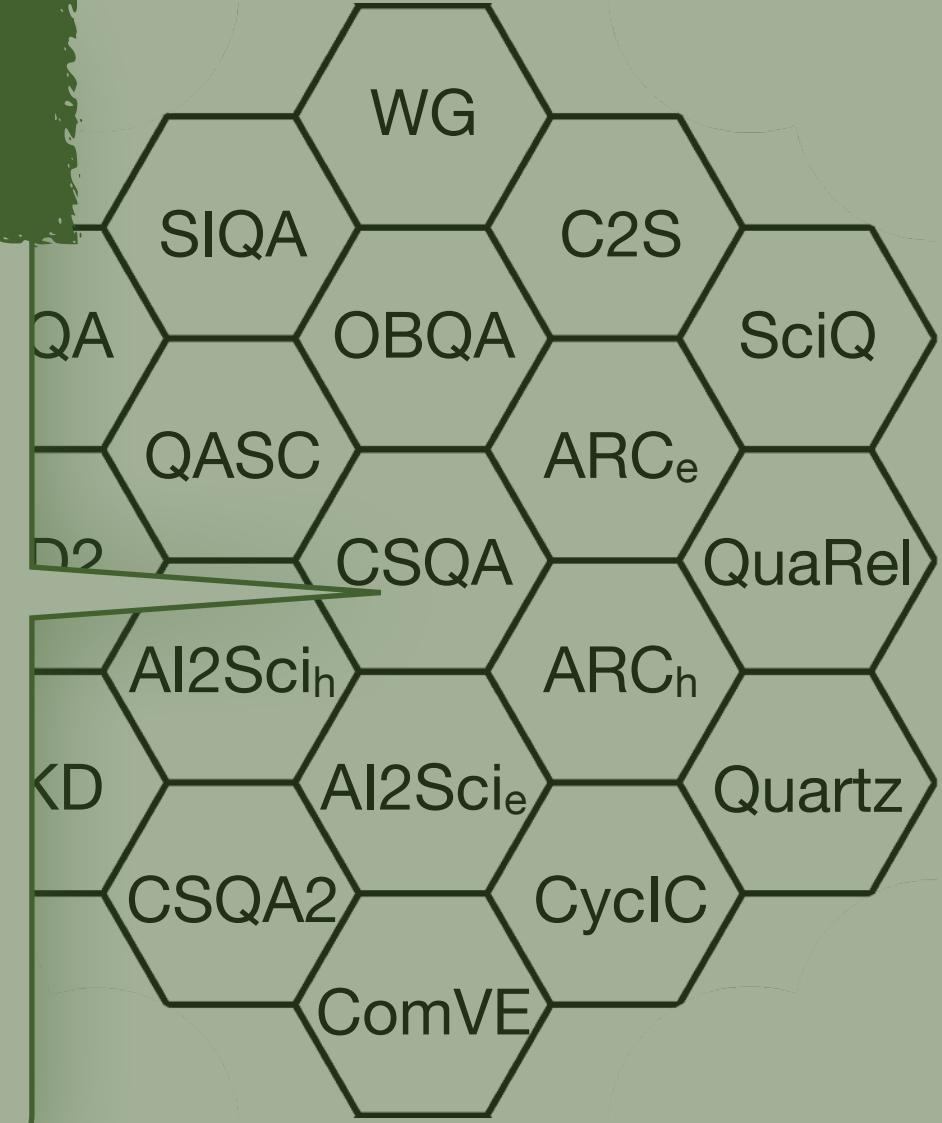
Trees live in cavities or hollows.

Trees grow using photosynthesis, absorbing carbon dioxide and releasing oxygen.

2 Knowledge Bases  
19 QA datasets  
~7M statements

Data Conversion

datasets



# Solving Commonsense Benchmarks

Predicting the most plausible statement out of the multiple-choice candidates

Name	Domain	Format
STAGE B TRAINING (SEEN)		
OpenBookQA	scientific	multiple-choice (4)
ARC (easy)	scientific	multiple-choice (4)
ARC (hard)	scientific	multiple-choice (4)

## Converted statement group:

Someone would wear an ungulate to protect themselves from a cannon.

(Incorrect)



3%

Someone would wear a bomber to protect themselves from a cannon.

(Incorrect)

6%

Someone would wear body armor to protect themselves from a cannon.

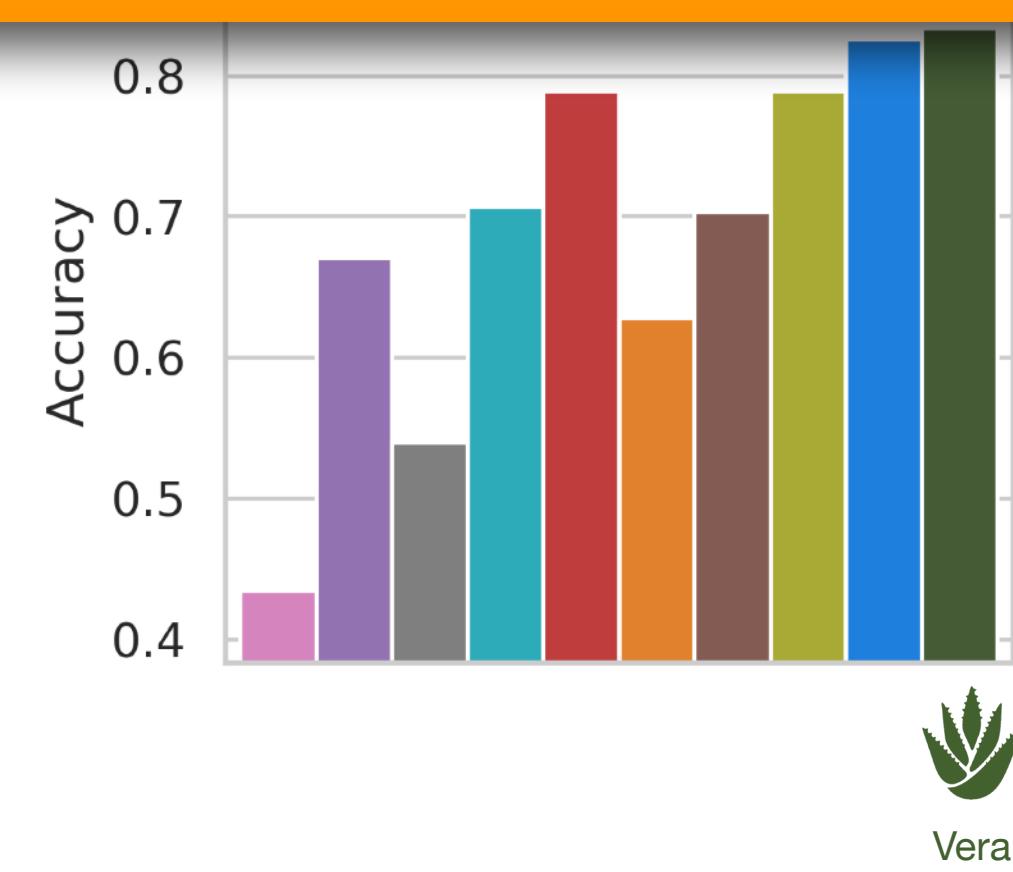
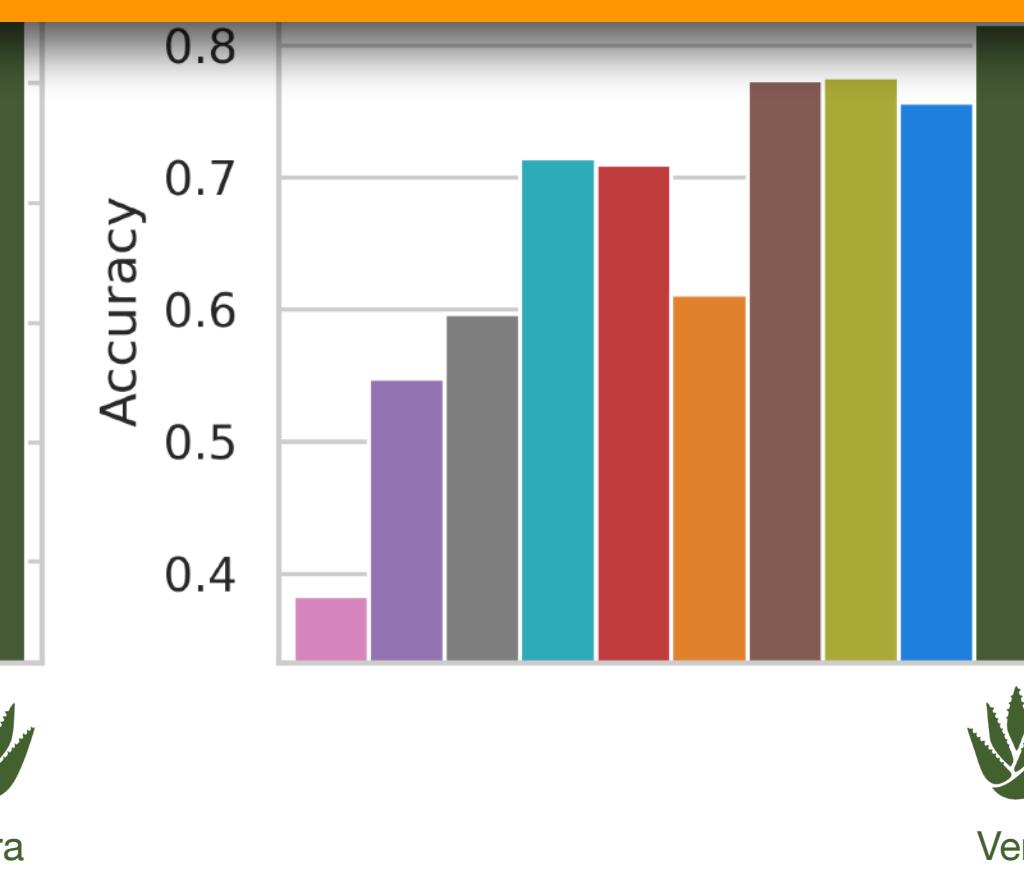
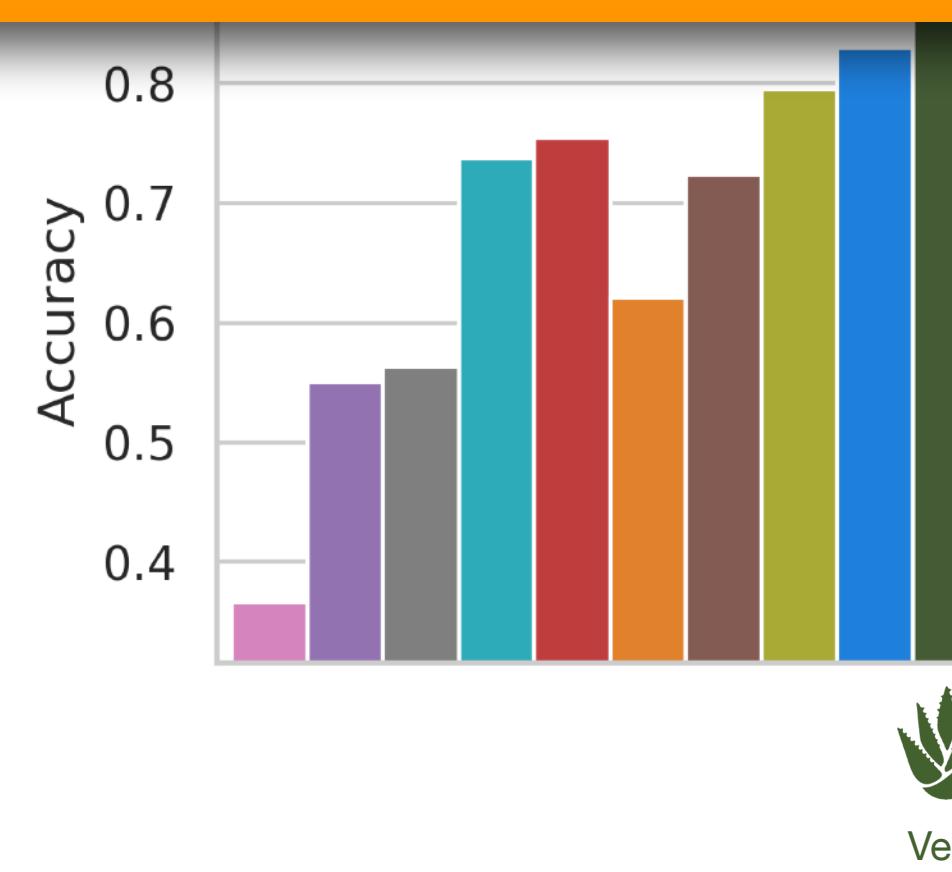
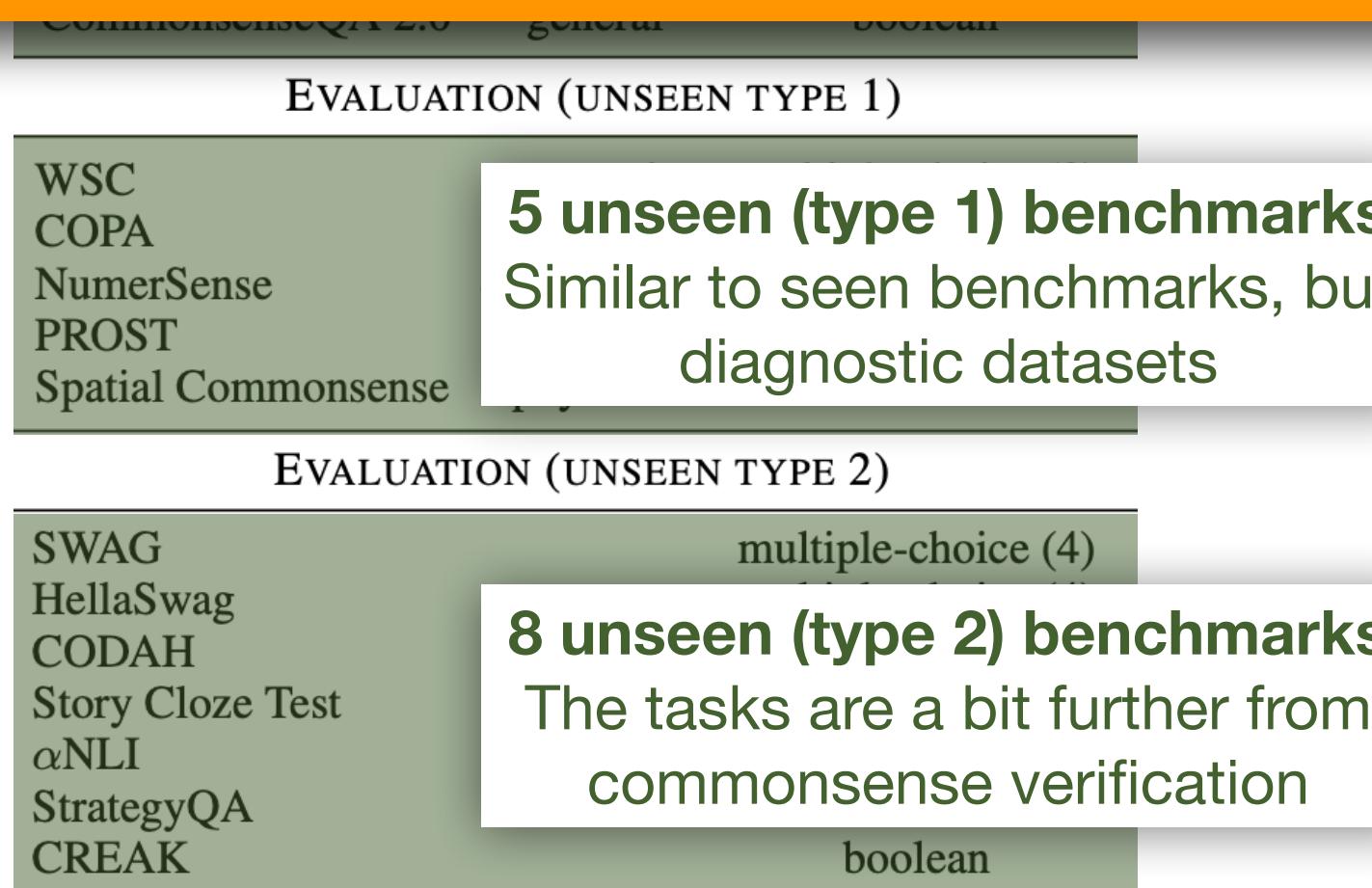
(Correct.)

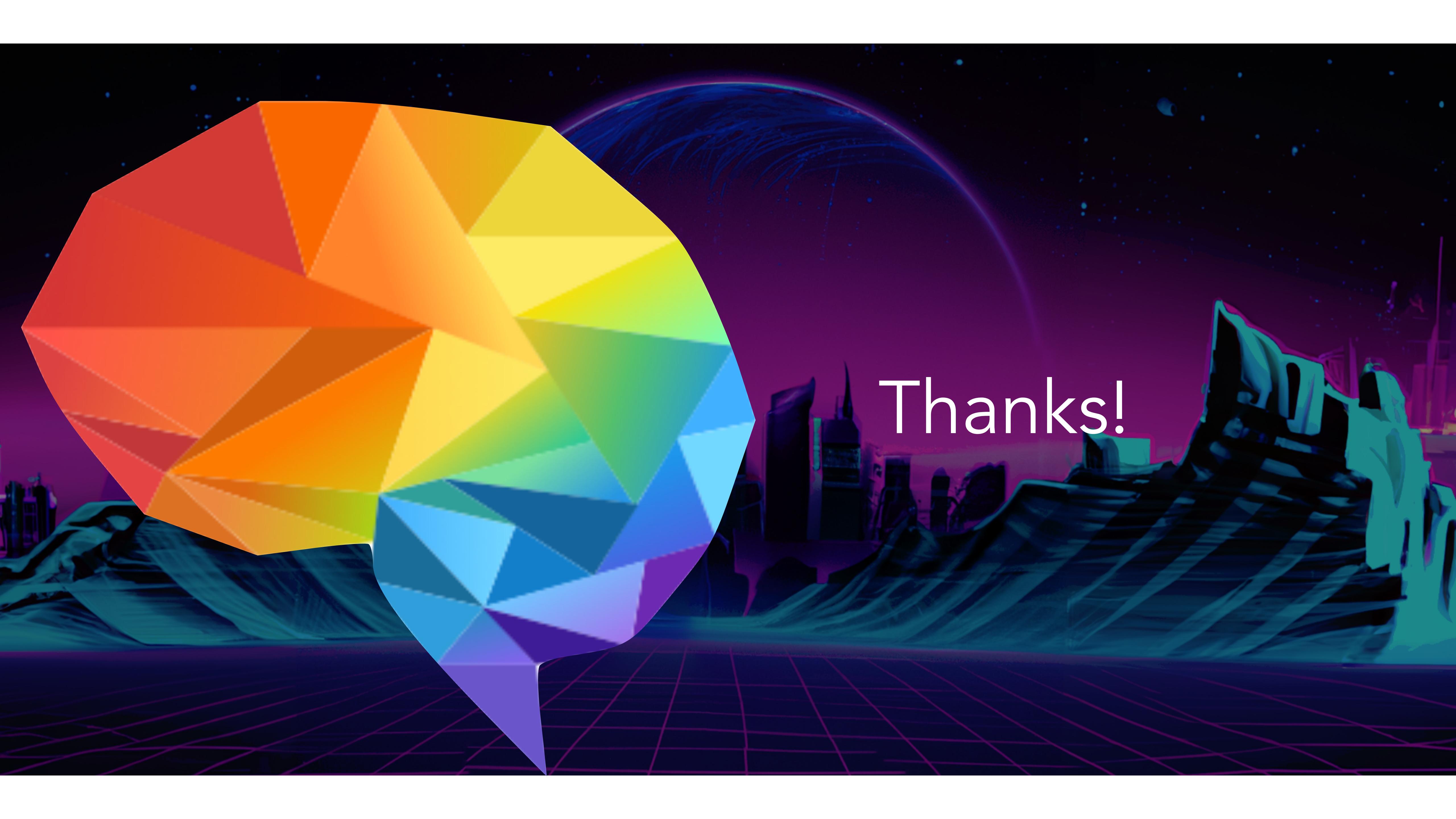
93%



Best baseline is Flan-T5. ChatGPT and GPT-4 are worse.

Vera outperforms Flan-T5 by 4%-6% on all eval sets (seen/unseen domains)





Thanks!