

2050: An AI Odyssey: Dark Matter of Intelligence

Yejin Choi

University of Washington & AI2

A futuristic cityscape at night, viewed from a low angle. The foreground is a dark, grid-patterned floor. In the middle ground, there are several buildings and structures, some of which are partially covered by large, dark, draped fabric. The background features a large, glowing planet or moon in the sky, surrounded by a starry night sky. The overall color palette is dominated by dark blues, purples, and greys, with a bright white glow from the planet and the text.

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 p

Diffusion generates

NeR

ChatGPT summarizes



like?

&

ormers?

aning?



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

ior:

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
f

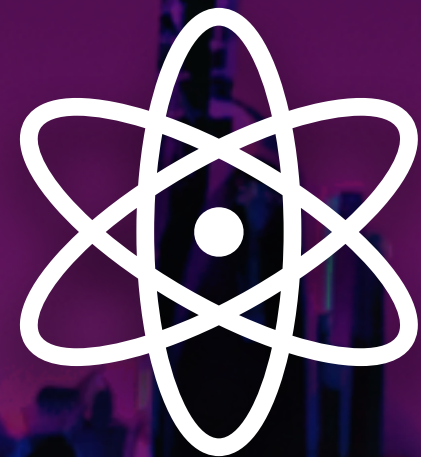
🔥 Quantum Physics 🔥

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,



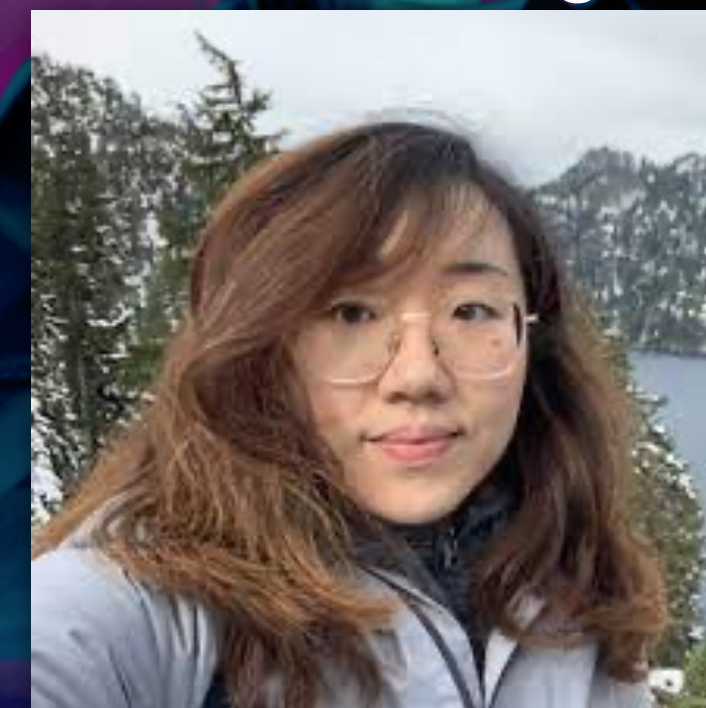
Melanie Sclar



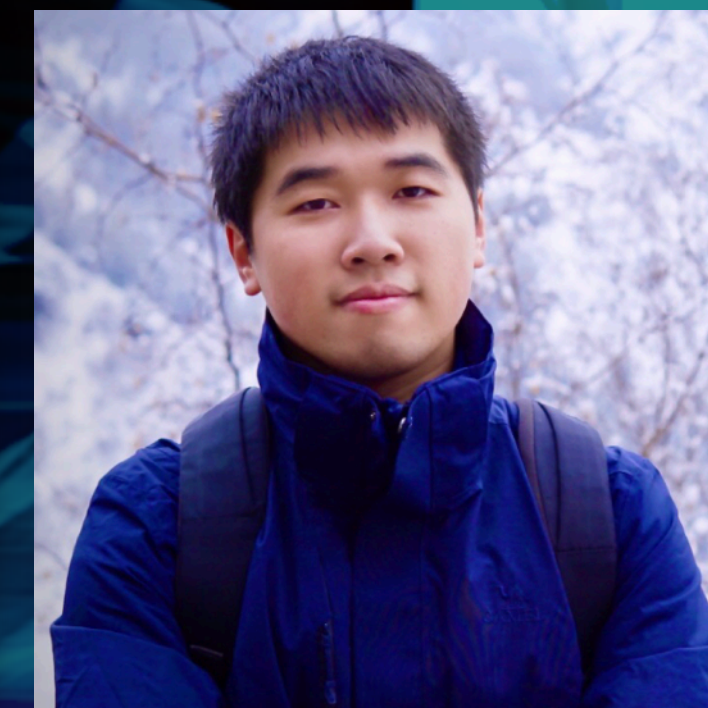
Xiang Lorraine Li,



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 Copy code
  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?



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 x 3-digit multiplication.

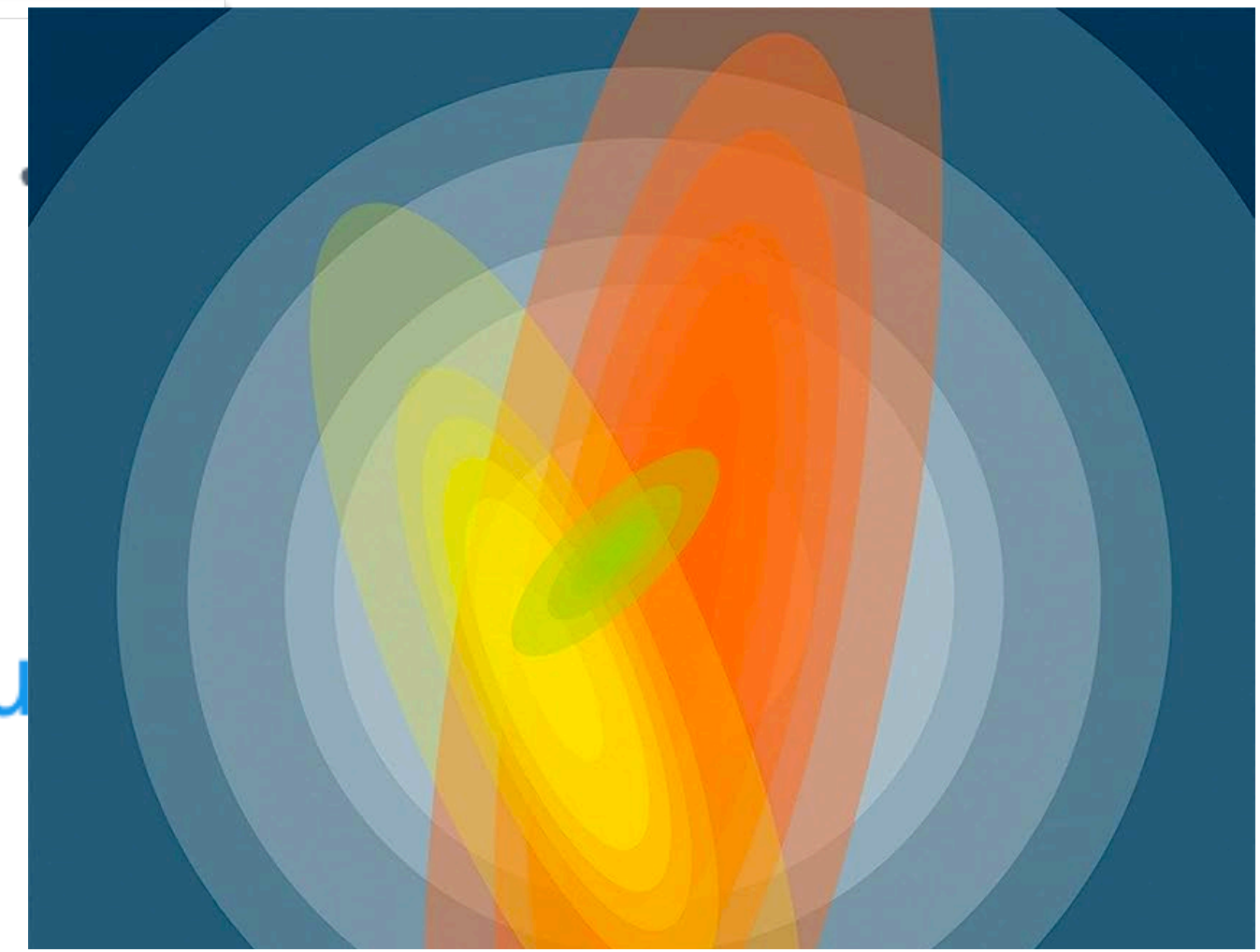
4% accuracy for 4x4 multiplication!

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 @ylecu 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



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)



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.

*1.8M examples include 1x1, 2x1, 2x2, 3x1, 3x2, 3x3, 4x1, 4x2 multiplication problem sizes, 12 epochs, 80:10:10 train/dev/test

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*)

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

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.



GPT4



Let's perform the multiplication step by step:

.....

The final answer is **16,695,134**.



Accuracy decreases to **4%**
on 4x4 multiplication!!

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.



GPT4



Let's perform the multiplication step by step:

.....

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



Accuracy decreases to **2%**
on 5x5 multiplication!!

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%

Partial Mastery



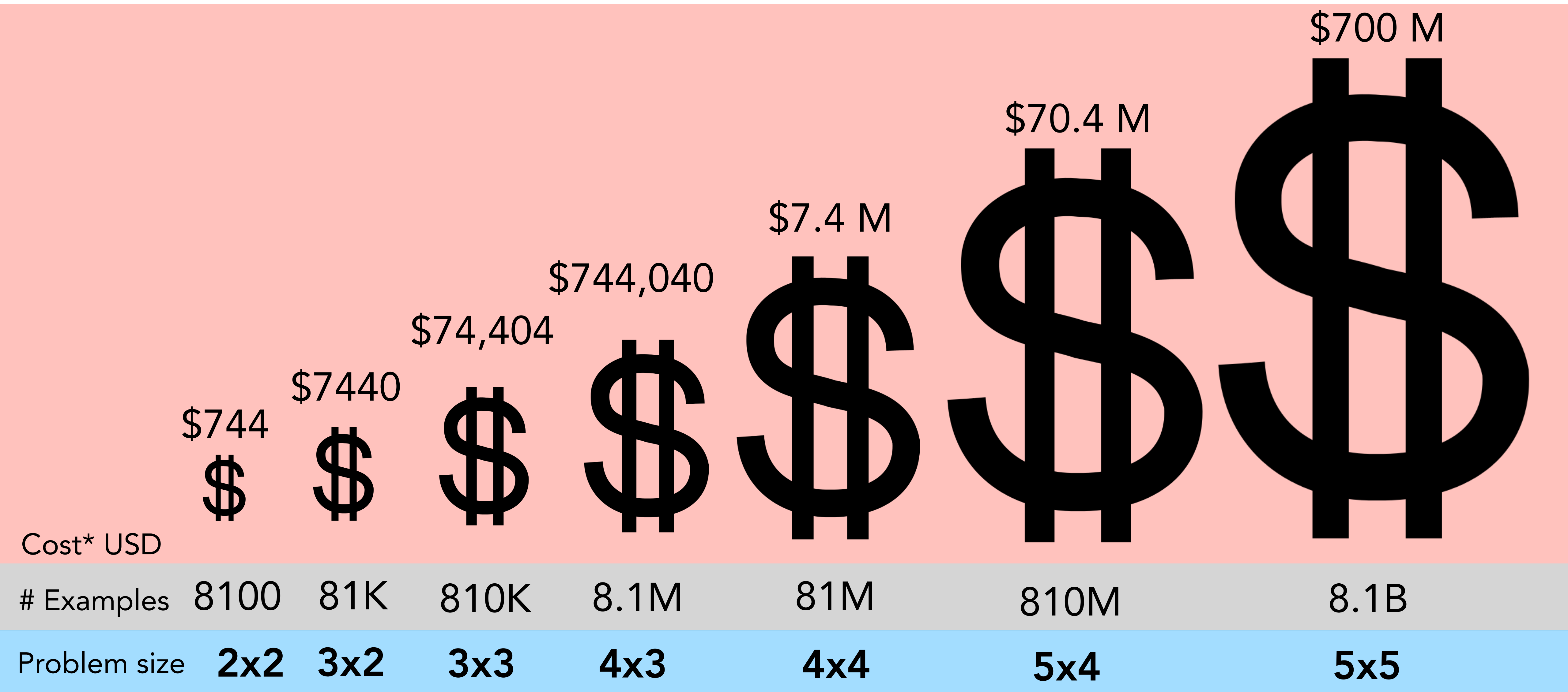
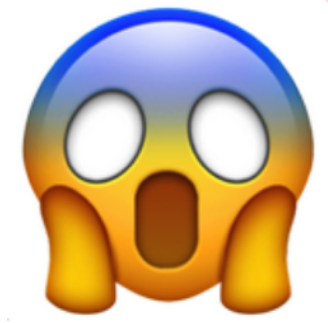
arxiv.org
Teaching Algorithmic Reasoning via In-context Learning

~~Instead~~

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

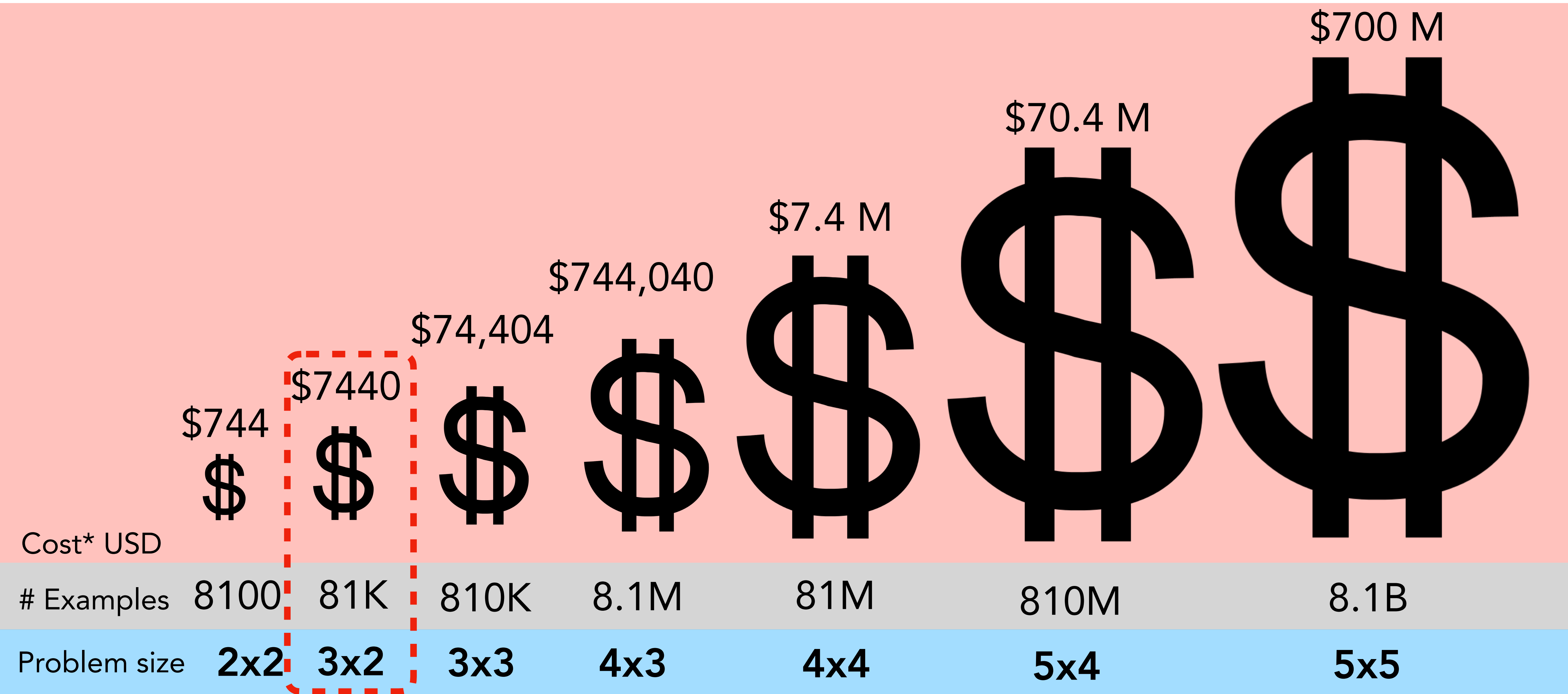
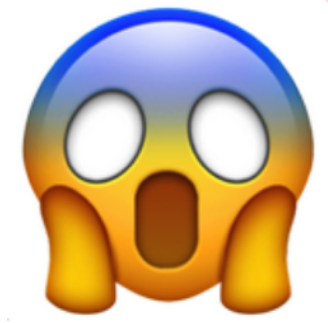
*they report GPT4 doesn't do well when multiplying digits > 3, thus covert the problem manually to addition over small digit (<= 3) multiplications

How about fine-tuning GPT3 on scratchpad? 🤖



*Cost for 4 epochs with text-davinci-003

How about fine-tuning GPT3 on scratchpad? 🤖



*Cost for 4 epochs with text-davinci-003

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**?*

Cost USD

Examples

\$744
\$7440

8100

81K

810K

8.1M

81M

810M

8.1B

\$70.4 M

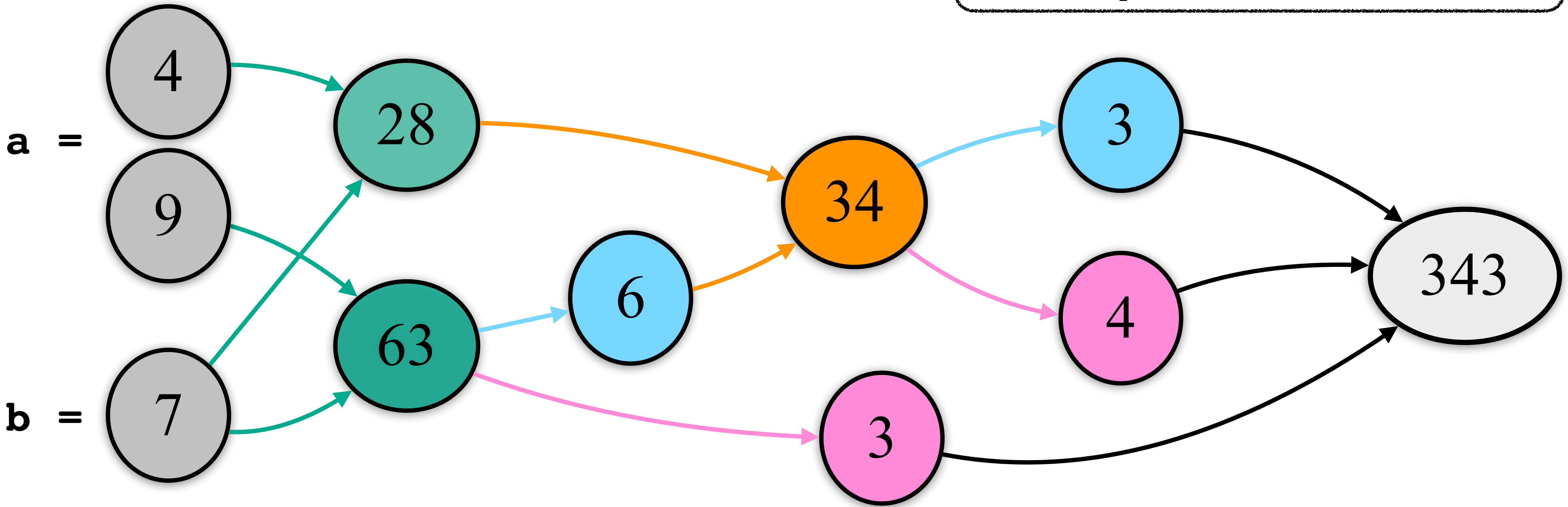
\$700 M

*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, 4x2, 4x4, etc

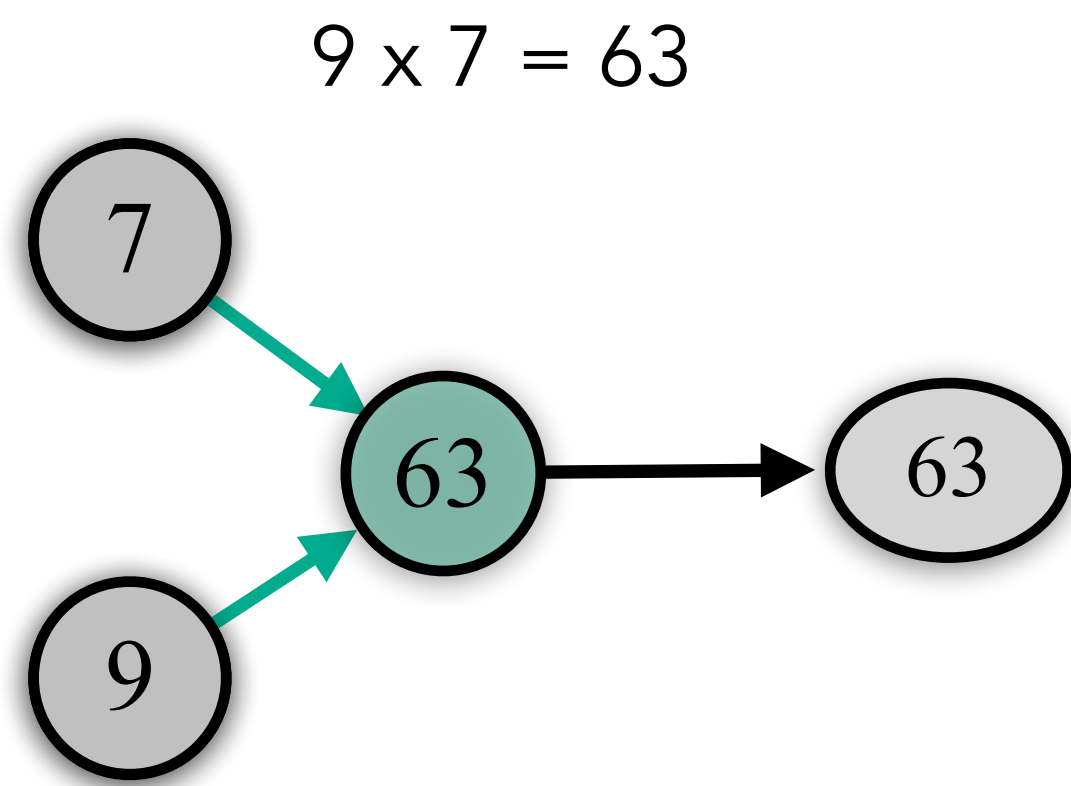
Computation graph for 49 x 7

```
function multiply (a[1:p], b[1:q]):
  for i = q to 1
    carry = 0
    for j = p to 1
      t = a[j] * b[i]
      t += carry (only if j != p)
      digits[j] = t mod 10
      carry = t // 10
    summands[i] = digits

  product =  $\sum_{i=1}^q \text{summands}[q+1-i] \cdot 10^{i-1}$ 
  return product
```



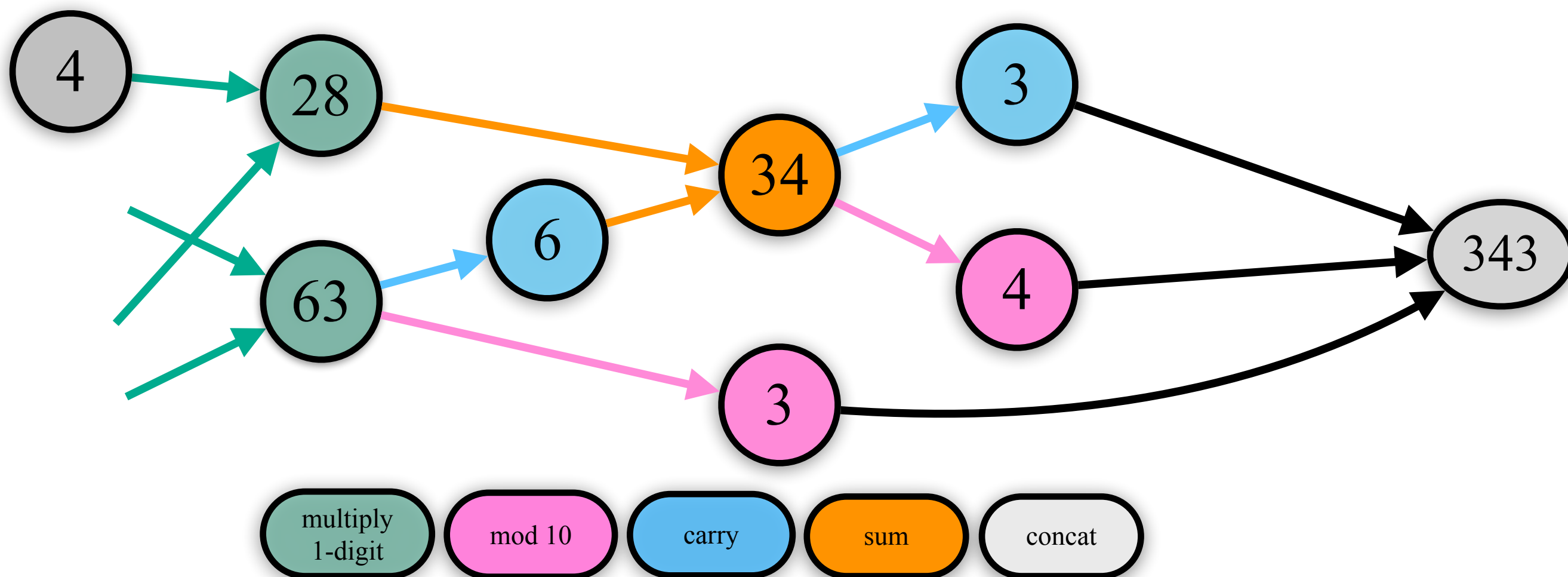
Model Performance Decreases as Graph Complexity Increases



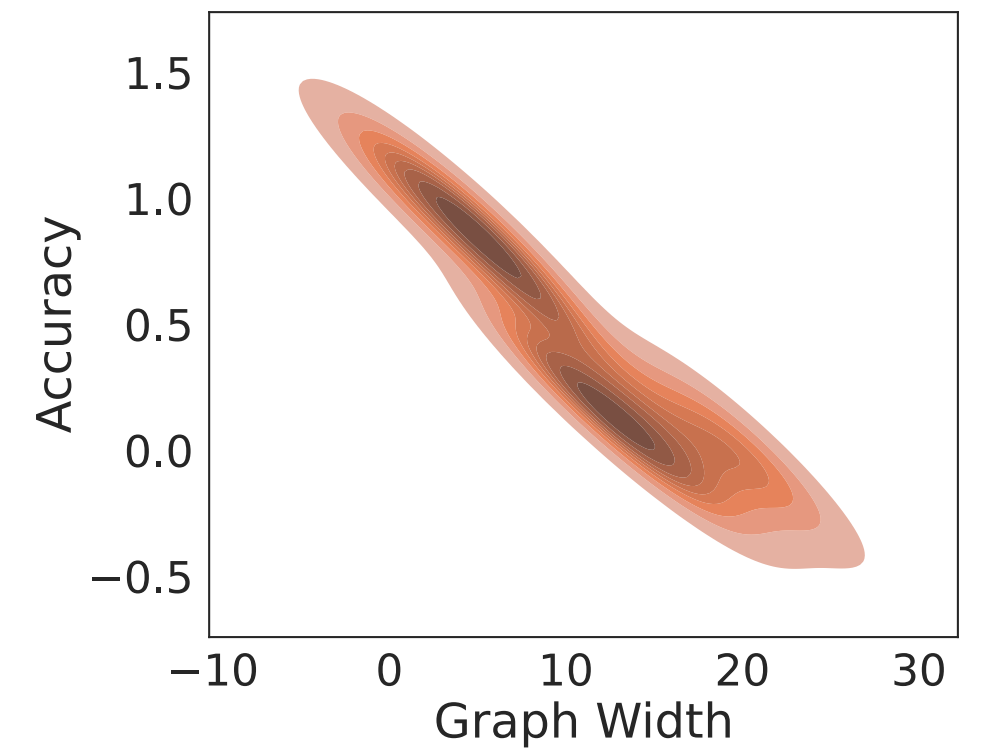
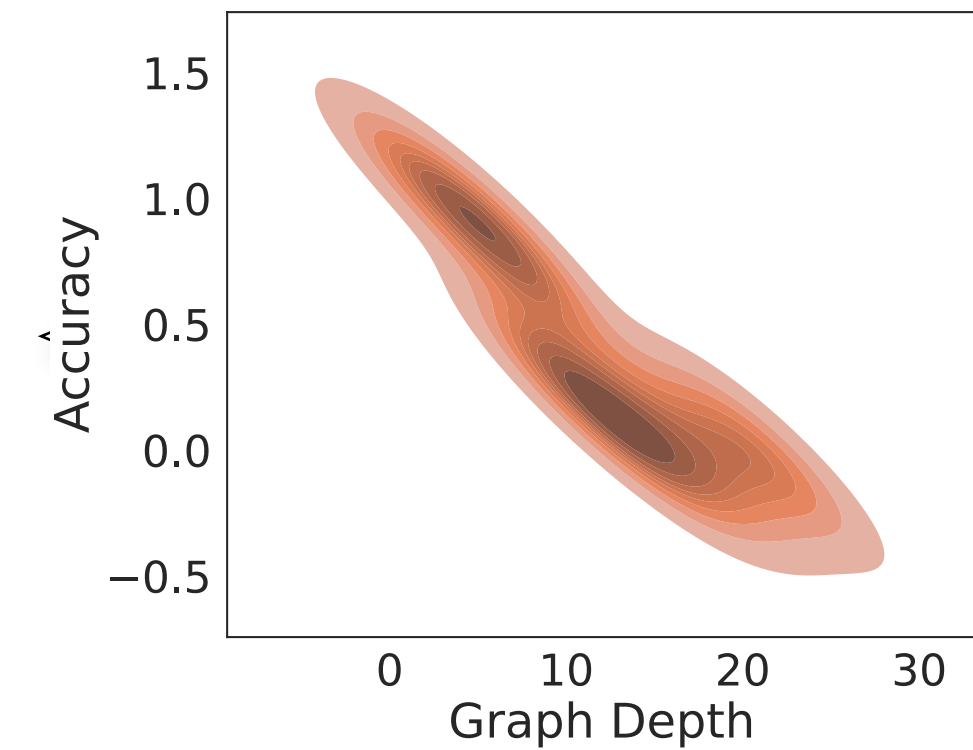
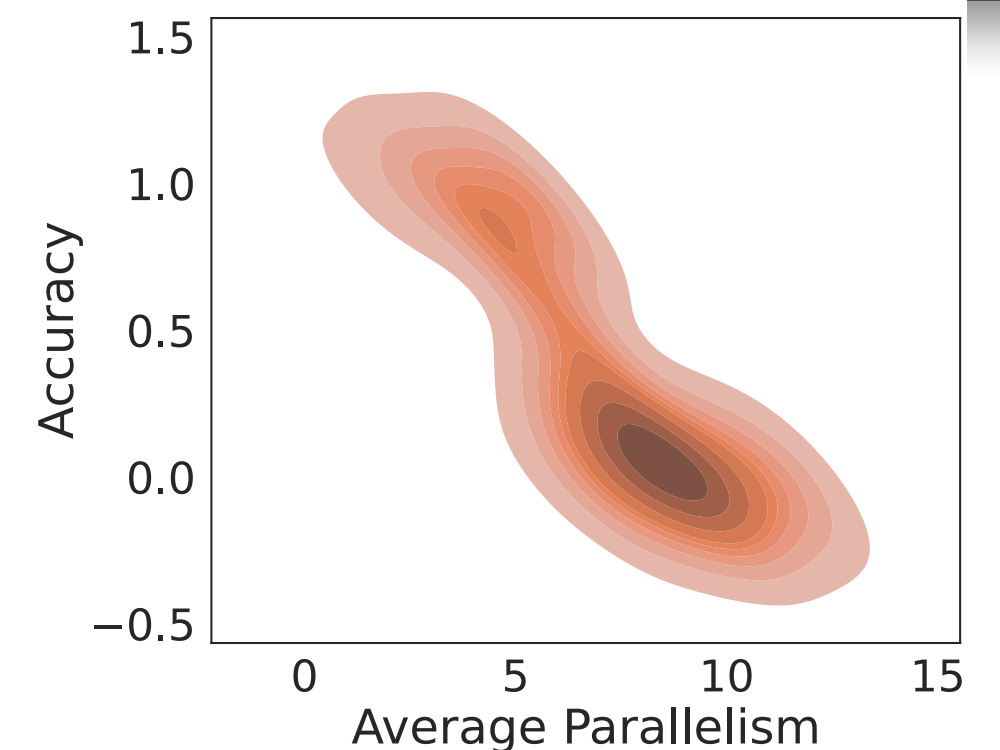
graph width = 1
 graph depth = 3
 avg. parallelism = 1.3

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

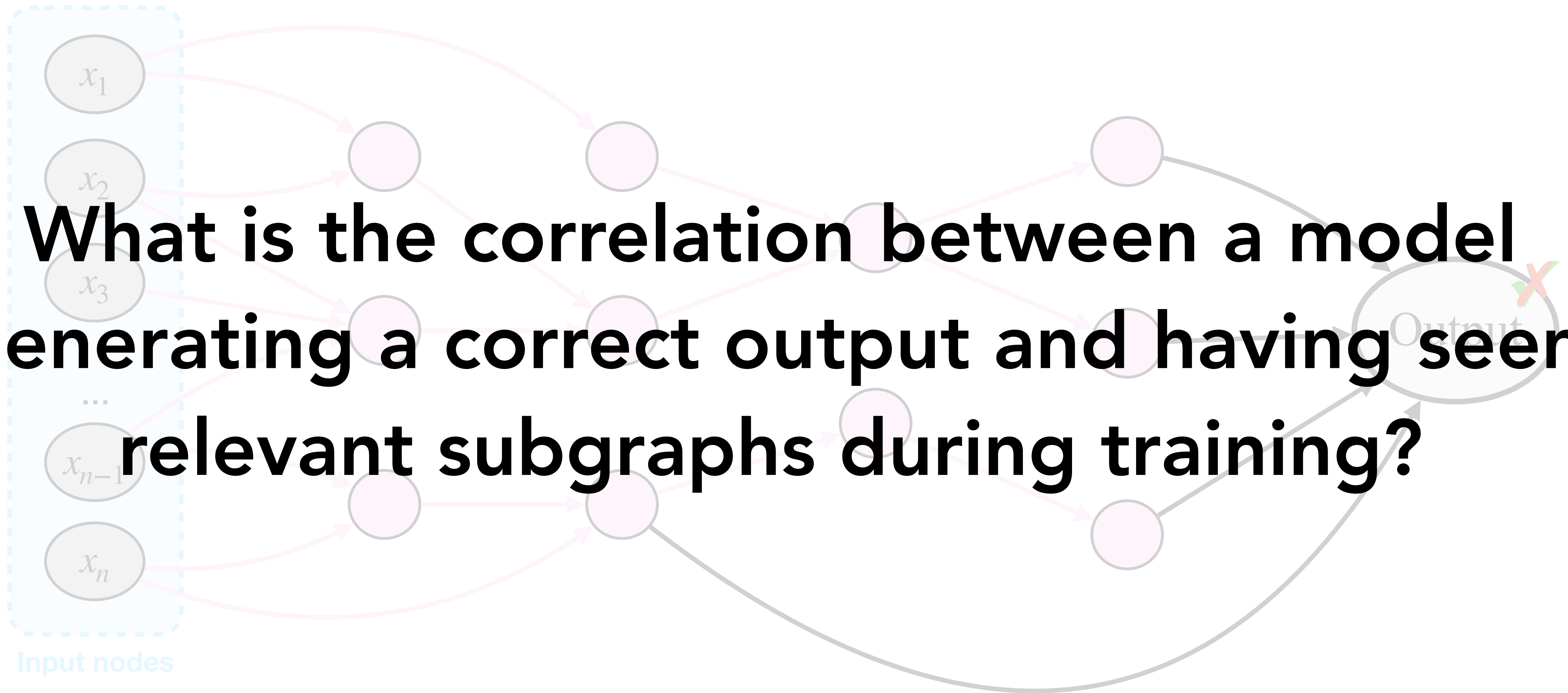
$49 \times 7 = 343$



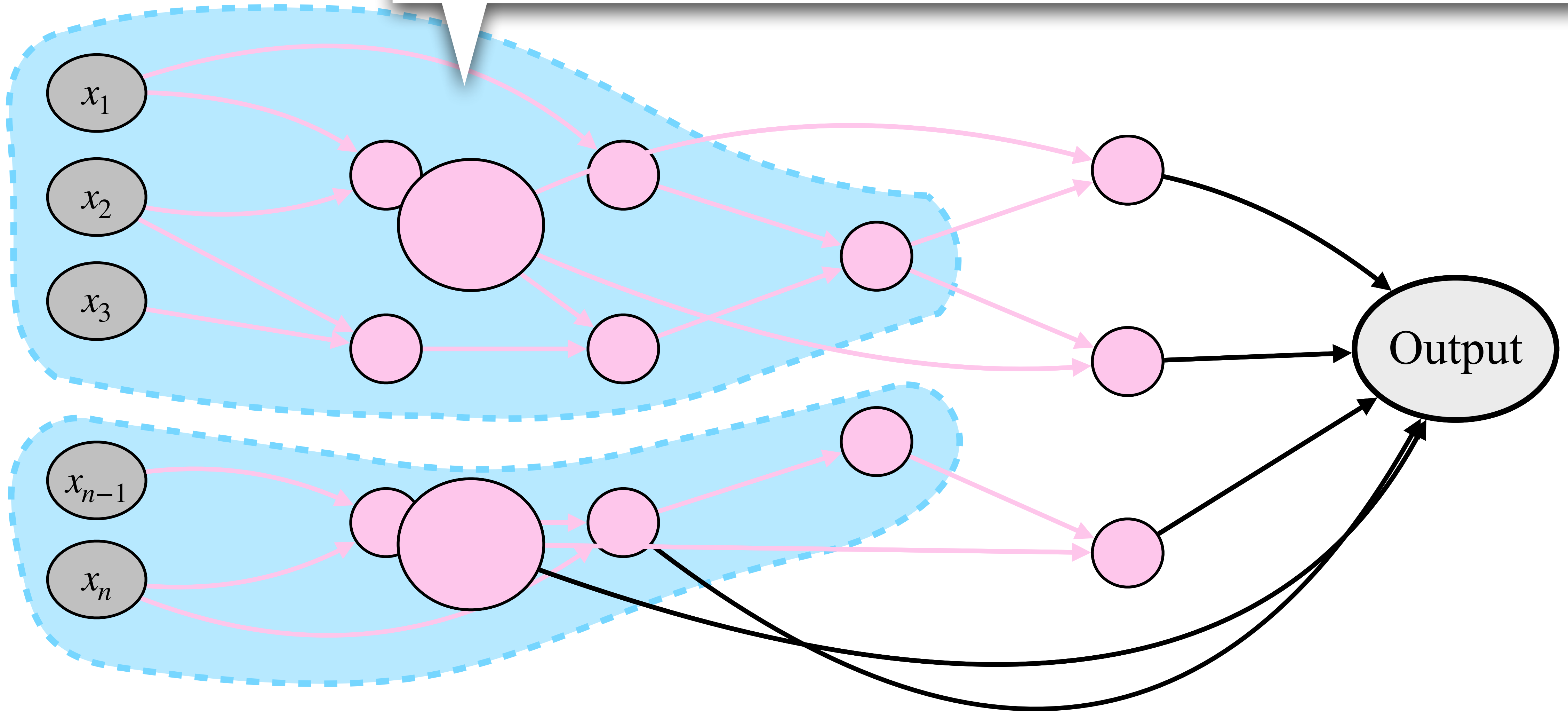
1	1				
2	1	0.99			
3	1	0.97	0.59		
4	0.96	0.78	0.23	0.04	
5	0.91	0.54	0.09	0.01	0
	1	2	3	4	5
	No. digits				



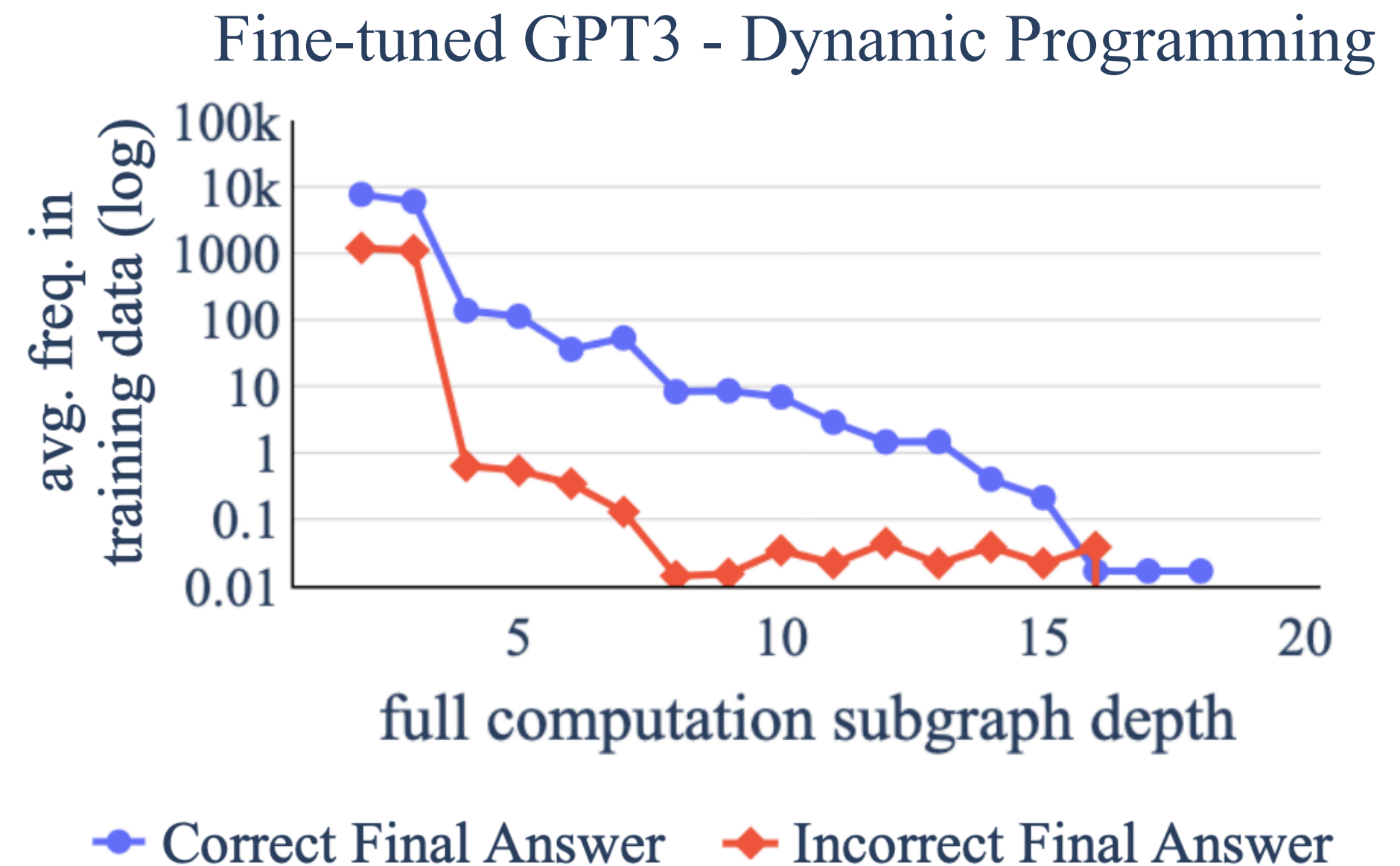
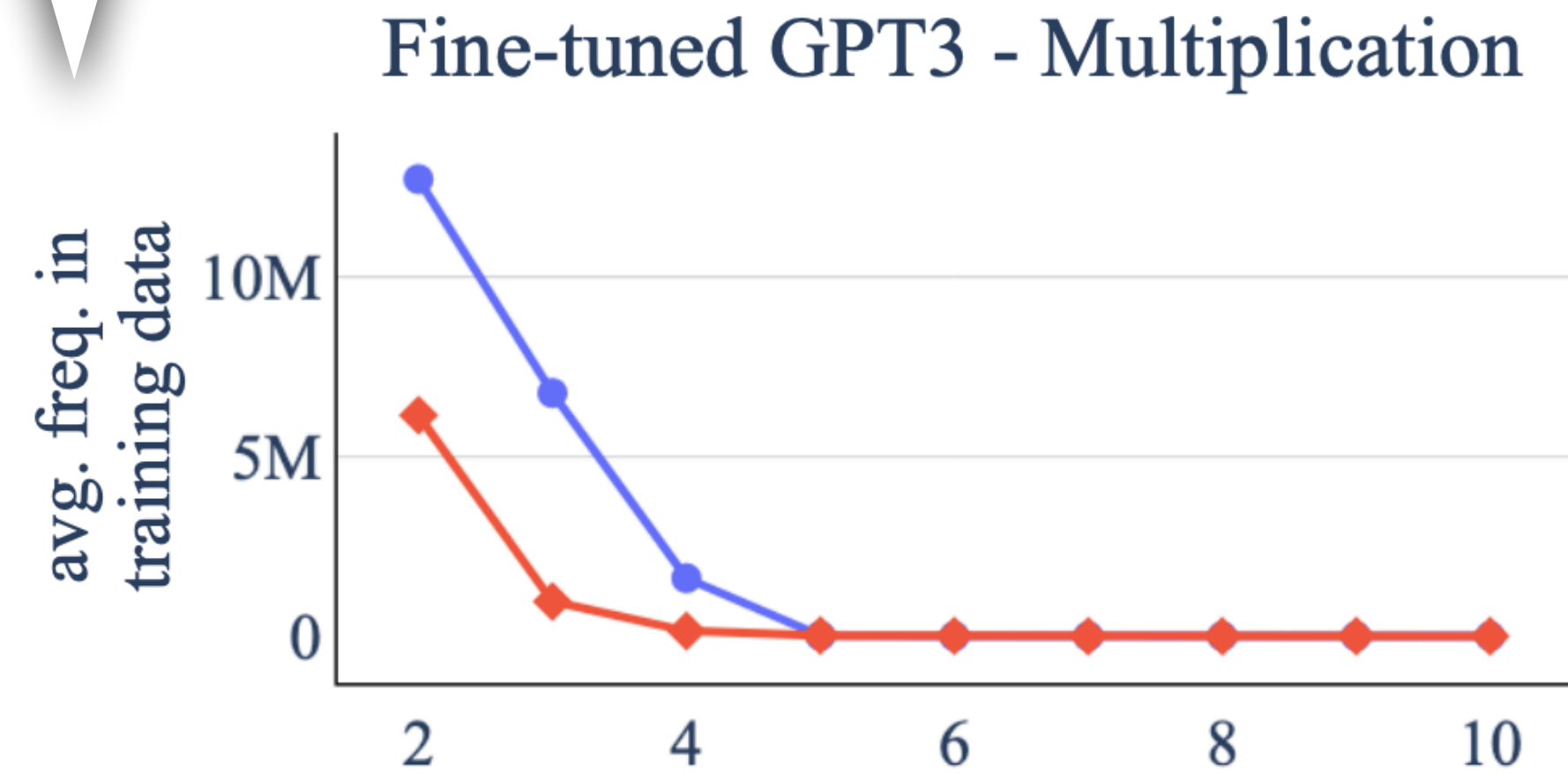
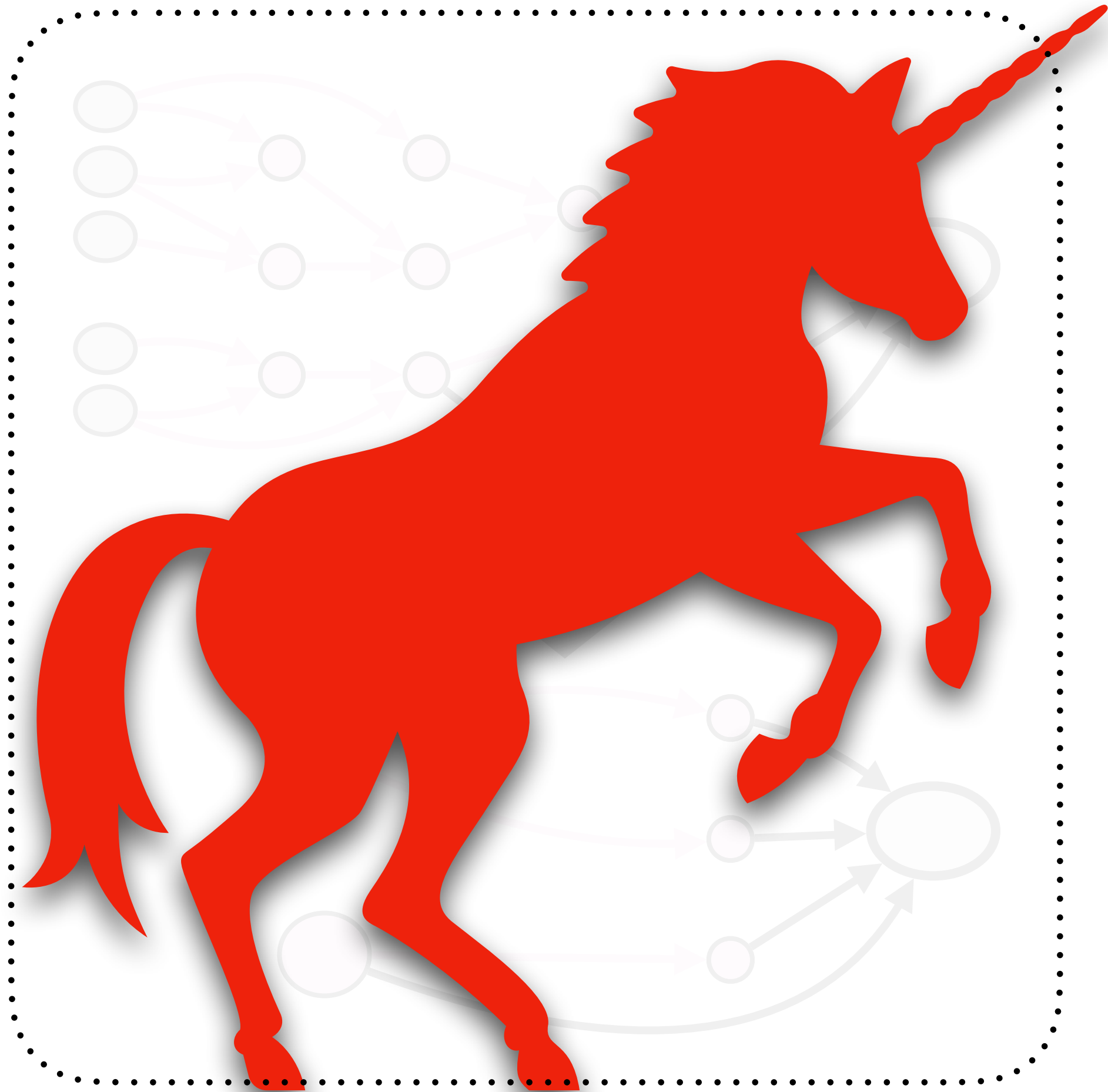
What is the correlation between a model generating a correct output and having seen relevant subgraphs during training?



Detect subgraphs already seen during training: *Identical* 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*



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

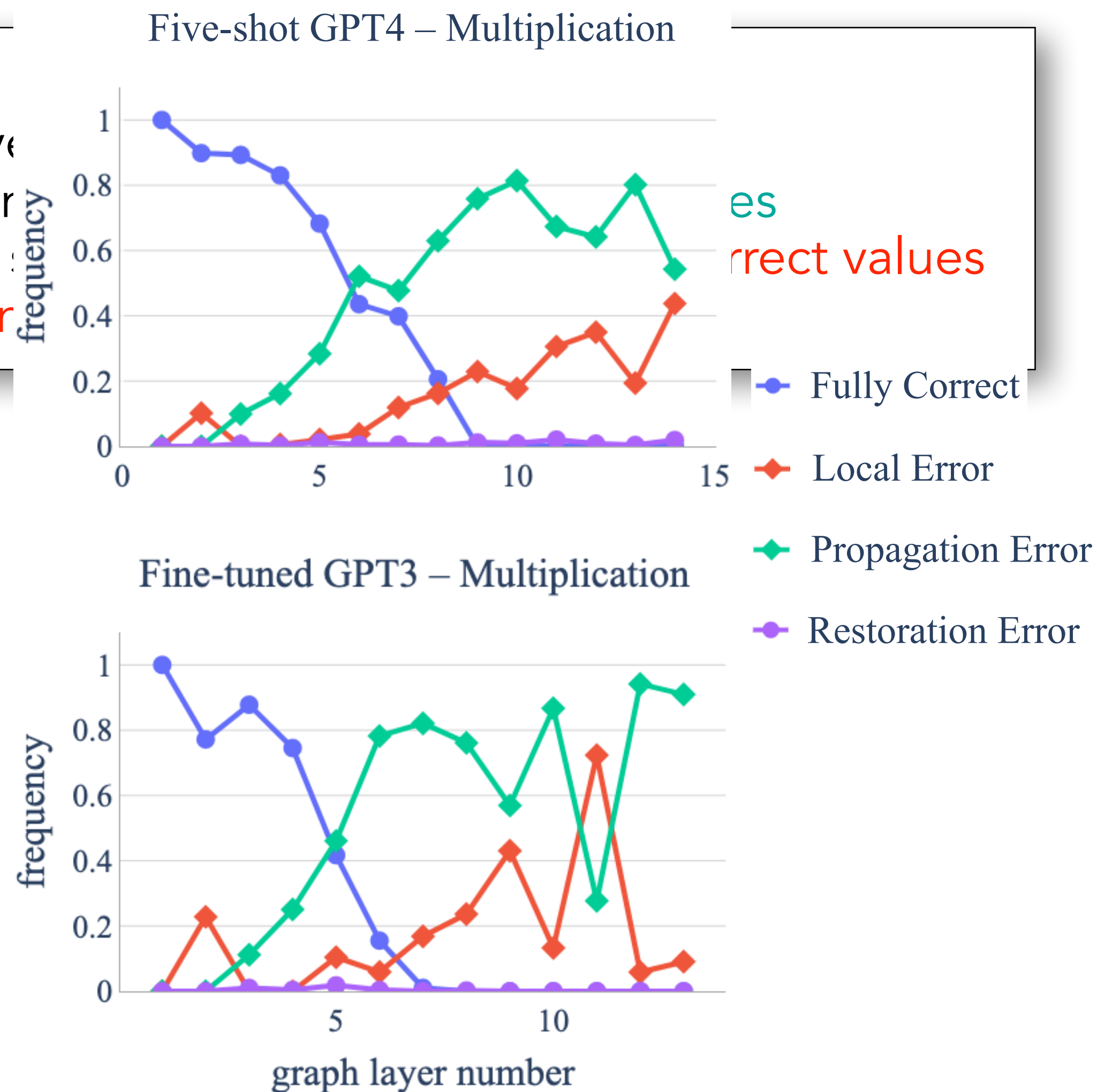
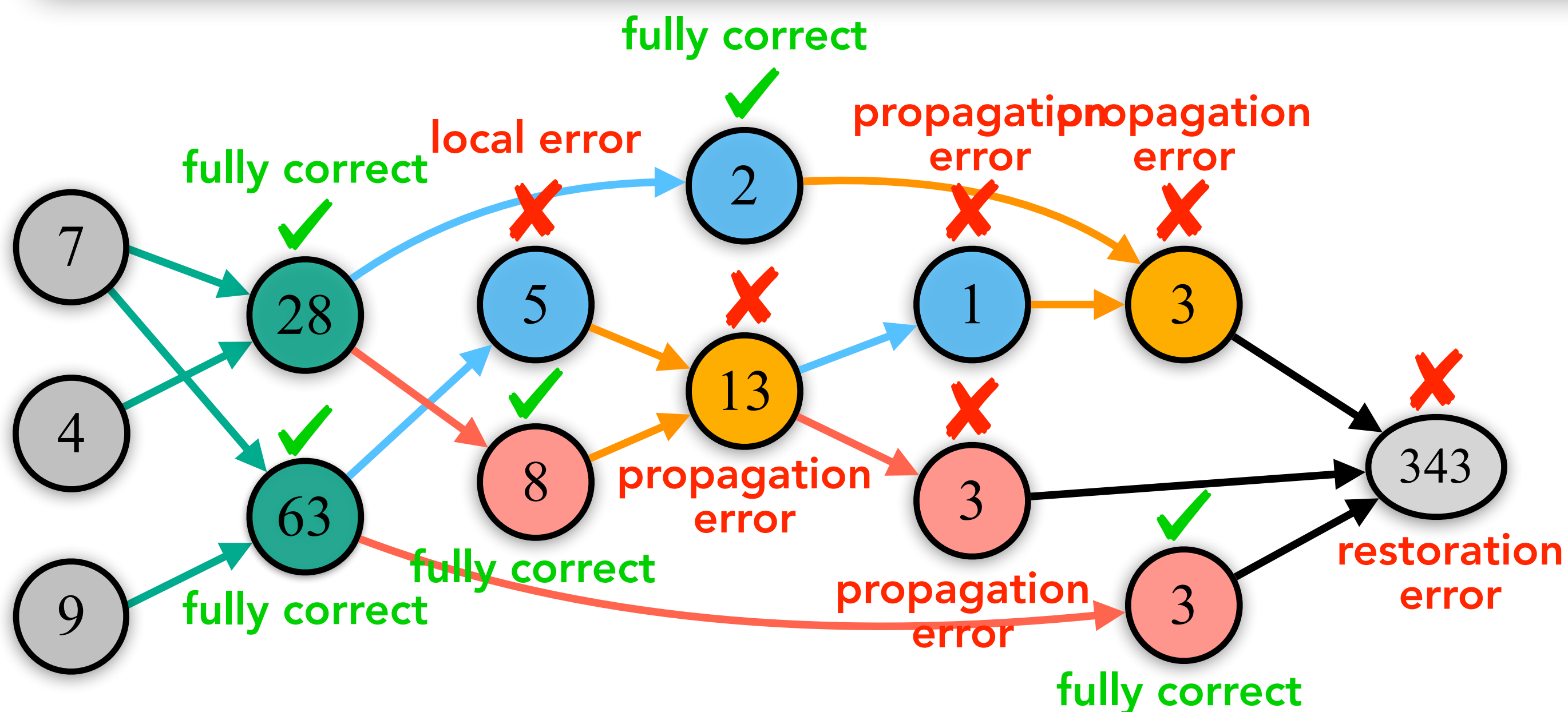
Error Type

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



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.

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

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

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

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 $\lim_{n \rightarrow +\infty} \mathbb{P}(f_n \neq \hat{f}_n) = 1$. \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.

$$s([x_1, \dots, x_m, x_{m+1}, \dots, x_{m+n}], j) := (\widehat{x_1 x_2 \dots x_m}, x_{m+j})$$

where $\widehat{x_1 x_2 \dots x_m}$ 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 ϵ and α 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.

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

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^{1,2,*,§}, Jörn-Henrik Jacobsen^{3,*}, Claudio Michaelis^{1,2,*},
Richard Zemel^{†,3}, Wieland Brendel^{†,1}, Matthias Bethge^{†,1} & Felix A. Wichmann^{†,1}

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[†], Dehan Kong[†], Longtao Huang[†], Hui Xue[†]

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 Sydney

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

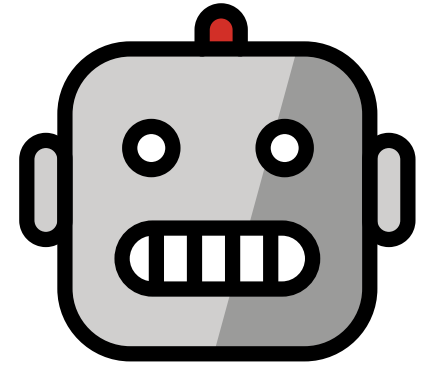


Yejin Choi

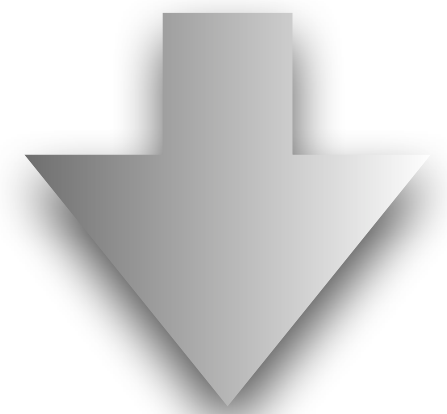


winning recipe = extreme-scale pre-training + RLHF at scale

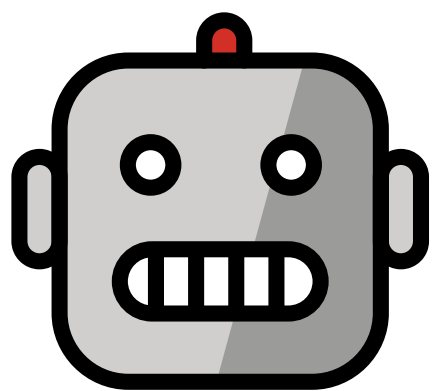
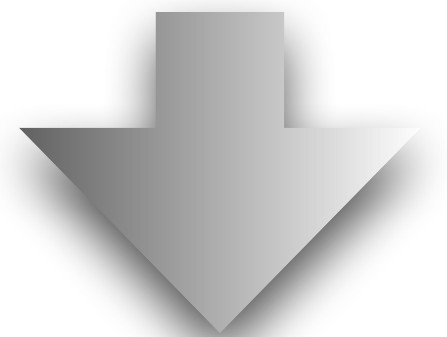
GPT-2



Low-quality, small models



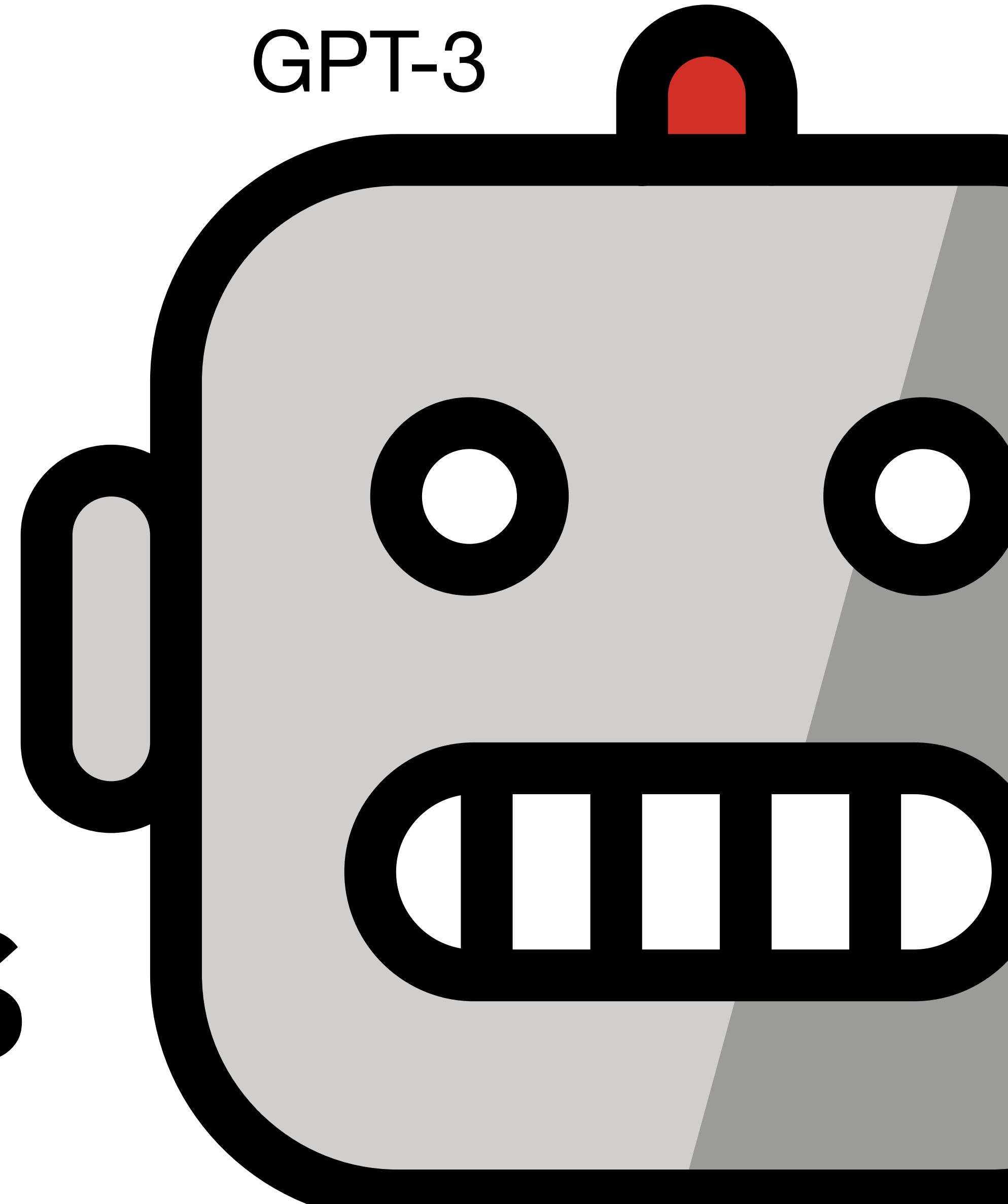
???



high-quality, small models

VS

GPT-3



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

Eric Wallace*
UC Berkeley
ericwallace@berkeley.edu

Charlie Snell*
UC Berkeley
csnell22@berkeley.edu

Xinyang Geng
UC Berkeley
young.geng@berkeley.edu

Hao Liu
UC Berkeley
hao.liu@berkeley.edu

Pieter Abbeel
UC Berkeley
pabbeel@berkeley.edu

Sergey Levine
UC Berkeley
svlevine@berkeley.edu

Dawn Song
UC Berkeley
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^{††*} Chandra Bhagavatula[‡] Jack Hessel[‡] Jena D. Hwang[‡]

John Bras[‡] Ximing Lu^{†‡} Sean Welleck^{†‡} Yejin Choi^{††*}
Computer Science & Engineering, University of Washington
Allen Institute for Artificial Intelligence

Teaching Small Language Models to Reason

Lucie Charlotte Magister^{*}

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

Alexander Severin

Google Research

severin@google.com

Specializing Smaller Language Models towards Multi-Step Reasoning

Yao Fu[♠] Hao Peng[♣] Litu Ou[♠] Ashish Sabharwal[♣] Tushar Khot[♣]

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

Cheng-Yu Hsieh^{1*}, Chun-Liang Li², Chih-Kuan Yeh³, Hootan Nakhost²,
Yasuhisa Fujii³, Alexander Ratner¹, Ranjay Krishna¹, Chen-Yu Lee², Tomas Pfister²

¹University of Washington, ²Google Cloud AI Research, ³Google 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” ✨

Mission Impossible:

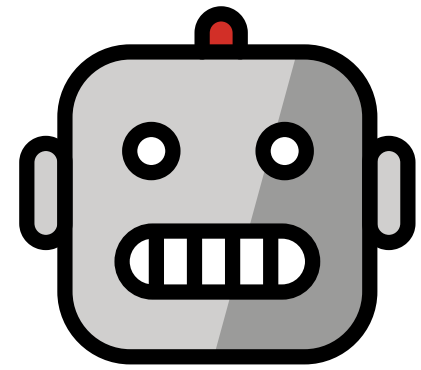
🔥 Learn to “summarize” + “paraphrase” 🔥

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

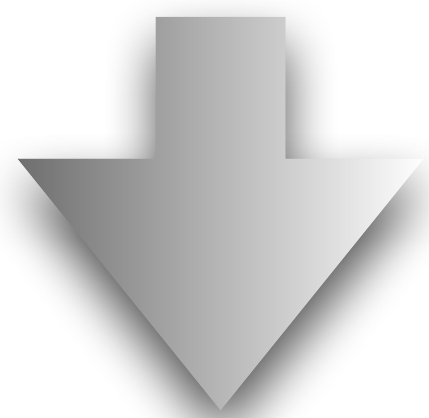
AI is as good as the data it was trained on

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

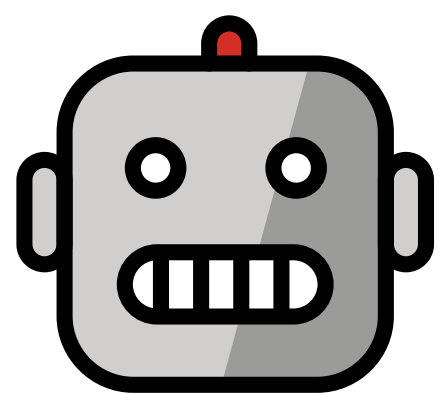
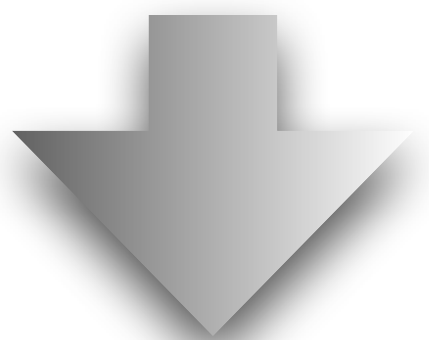
GPT-2



Low-quality, small models



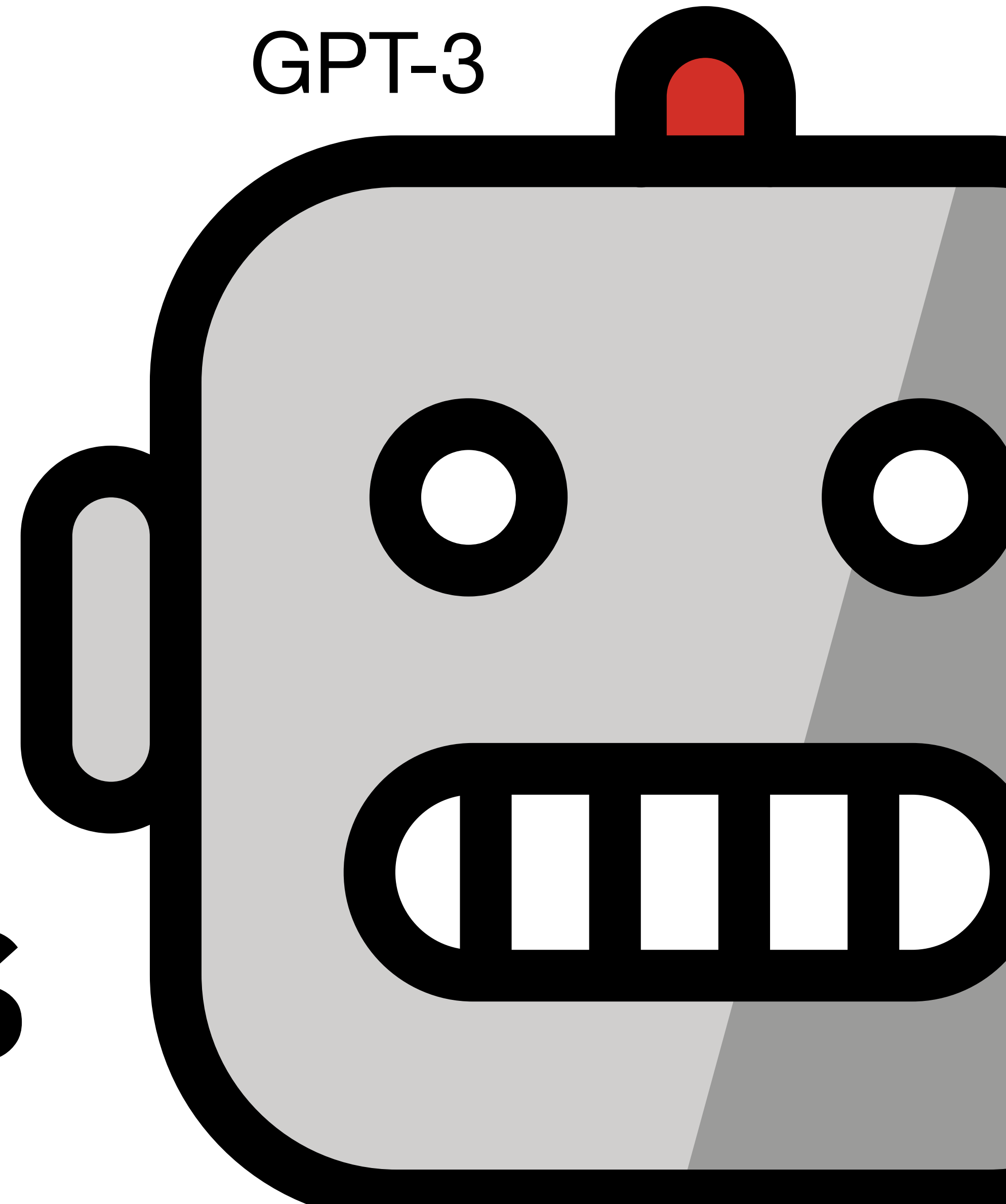
high-quality, large datasets



high-quality, small models

VS

GPT-3



We will build on ...

Symbolic Knowledge Distillation

From General Language Models to **Commonsense** Models

— NAACL 2022 —

New:

ATOMIC-10x

COMET-distill



Peter
West

Chandra
Bhagavatula



Jack
Hessel



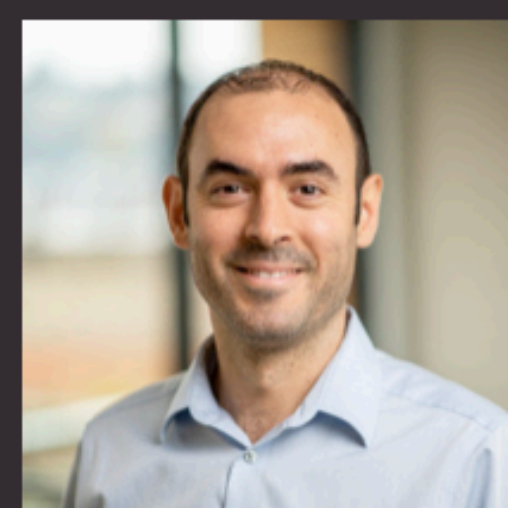
Jena
Hwang



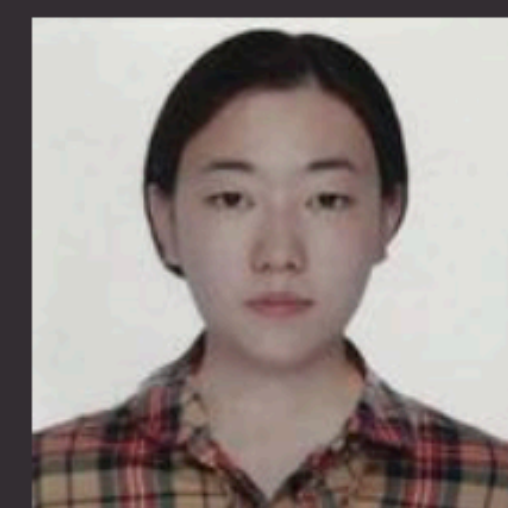
Liwei
Jiang



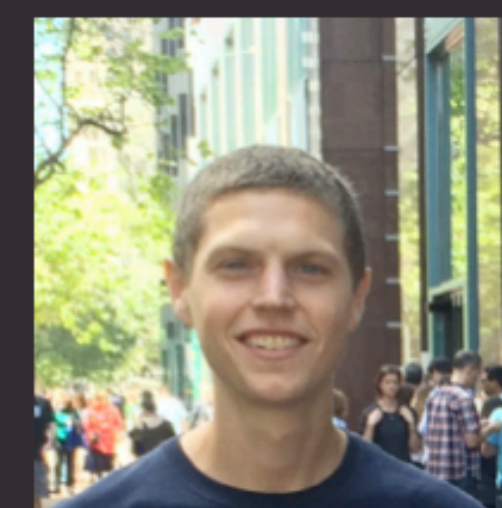
Ronan
Le Bras



Ximing
Lu



Sean
Welleck



Yejin
Choi



Symbolic Knowledge Distillation

From General Language Models to Commonsense Models

— NAACL 2022 —

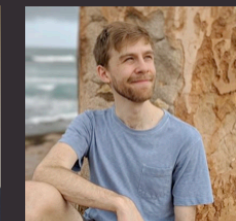


Peter West

Chandra Bhagavatula



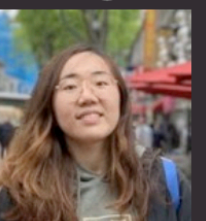
Jack Hessel



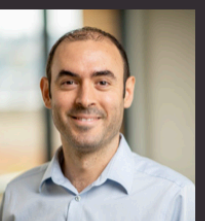
Jena Hwang



Liwei Jiang



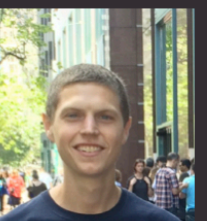
Ronan Le Bras



Ximing Lu



Sean Welleck



Yejin Choi

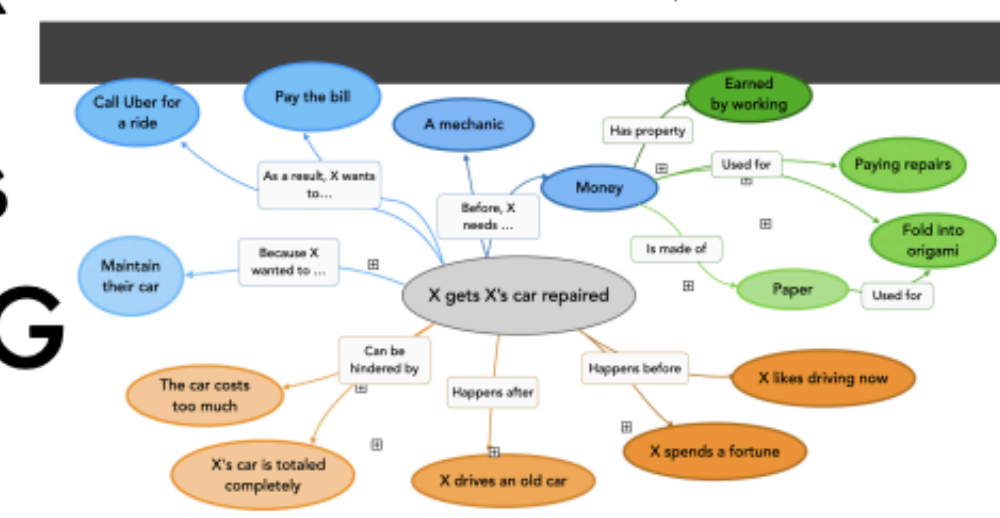
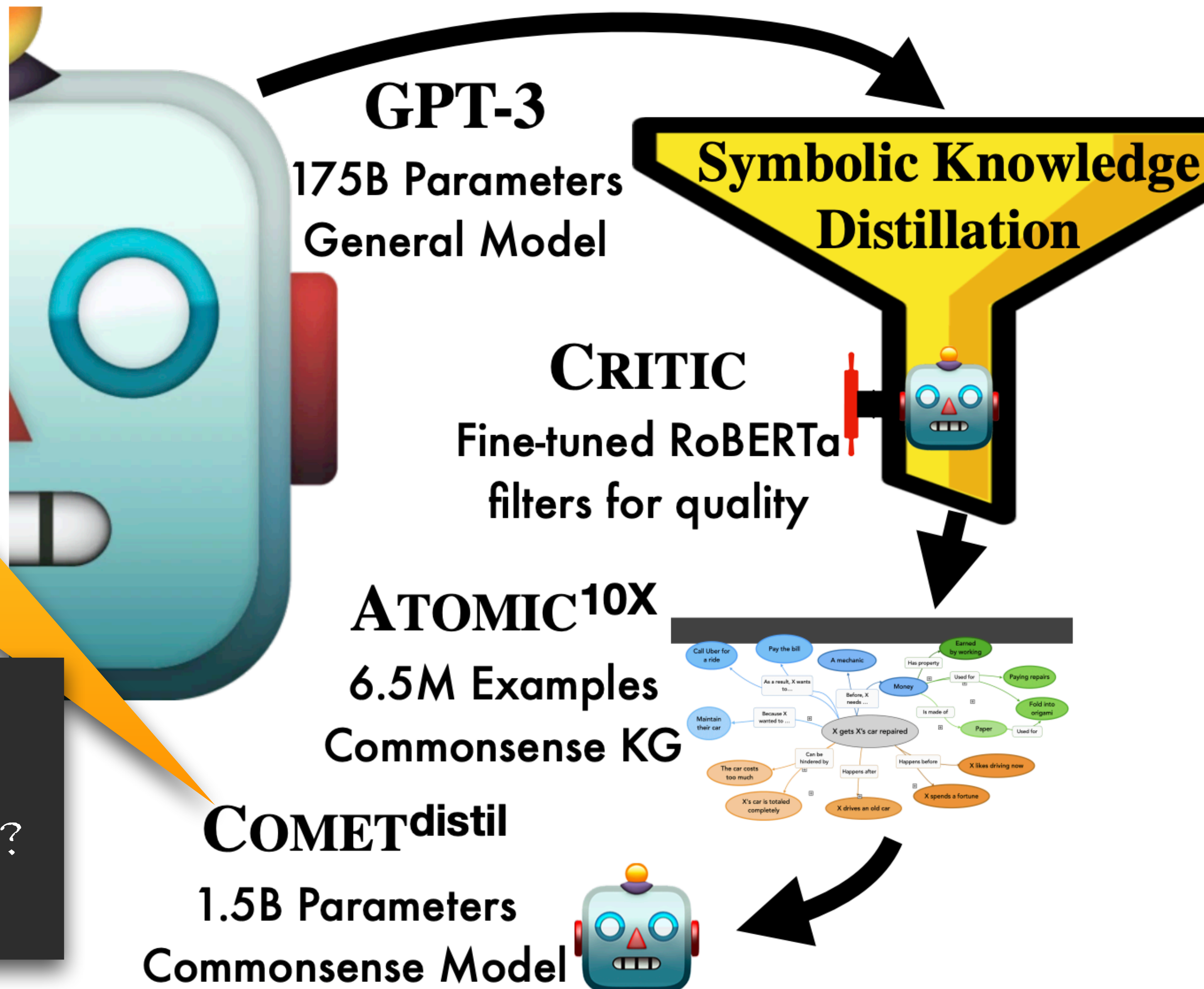


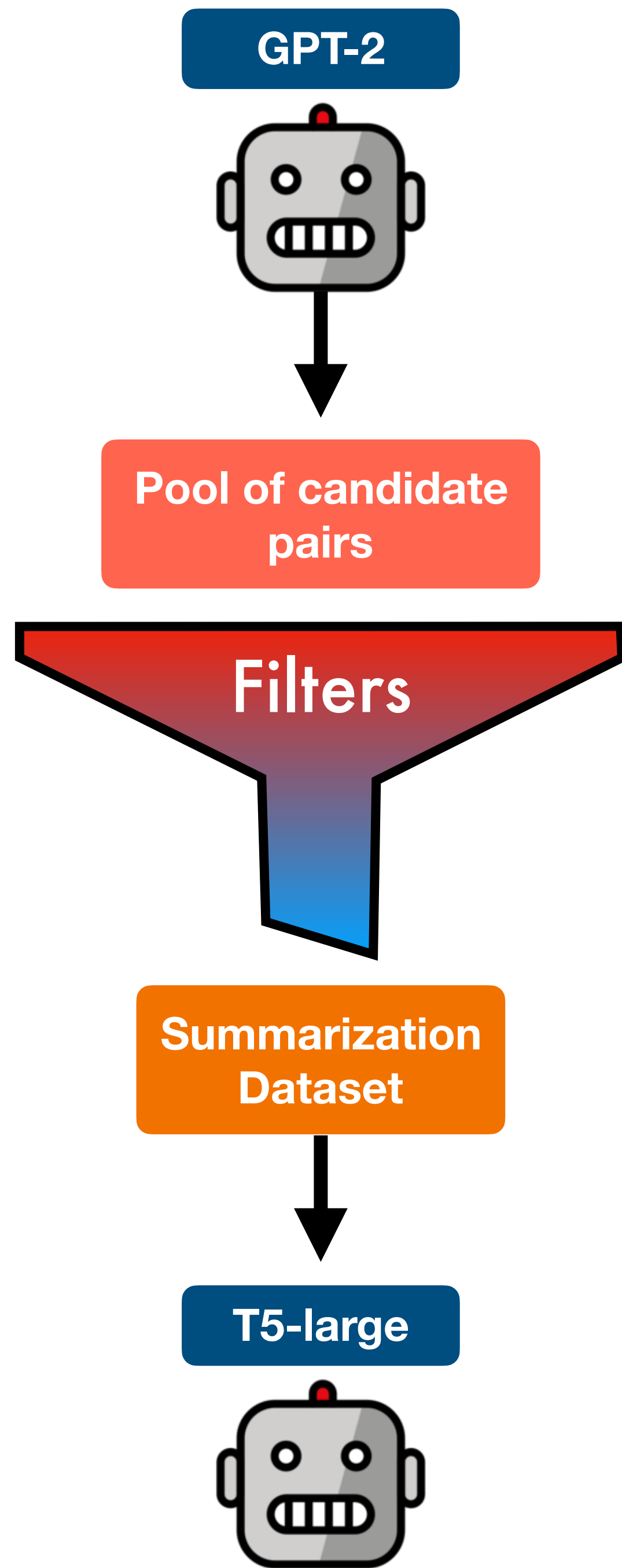
New:
ATOMIC-10x
COMET-distill

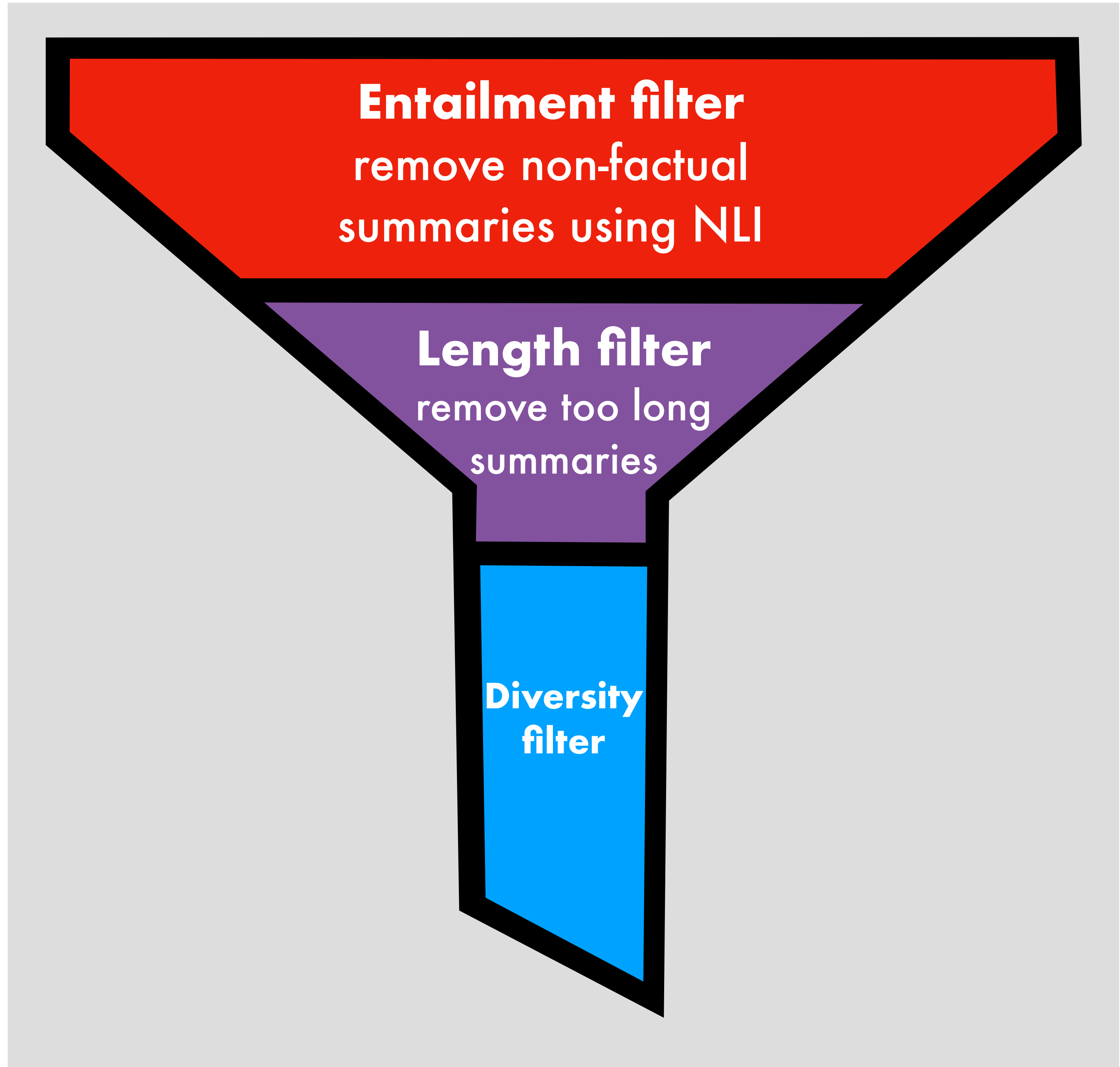
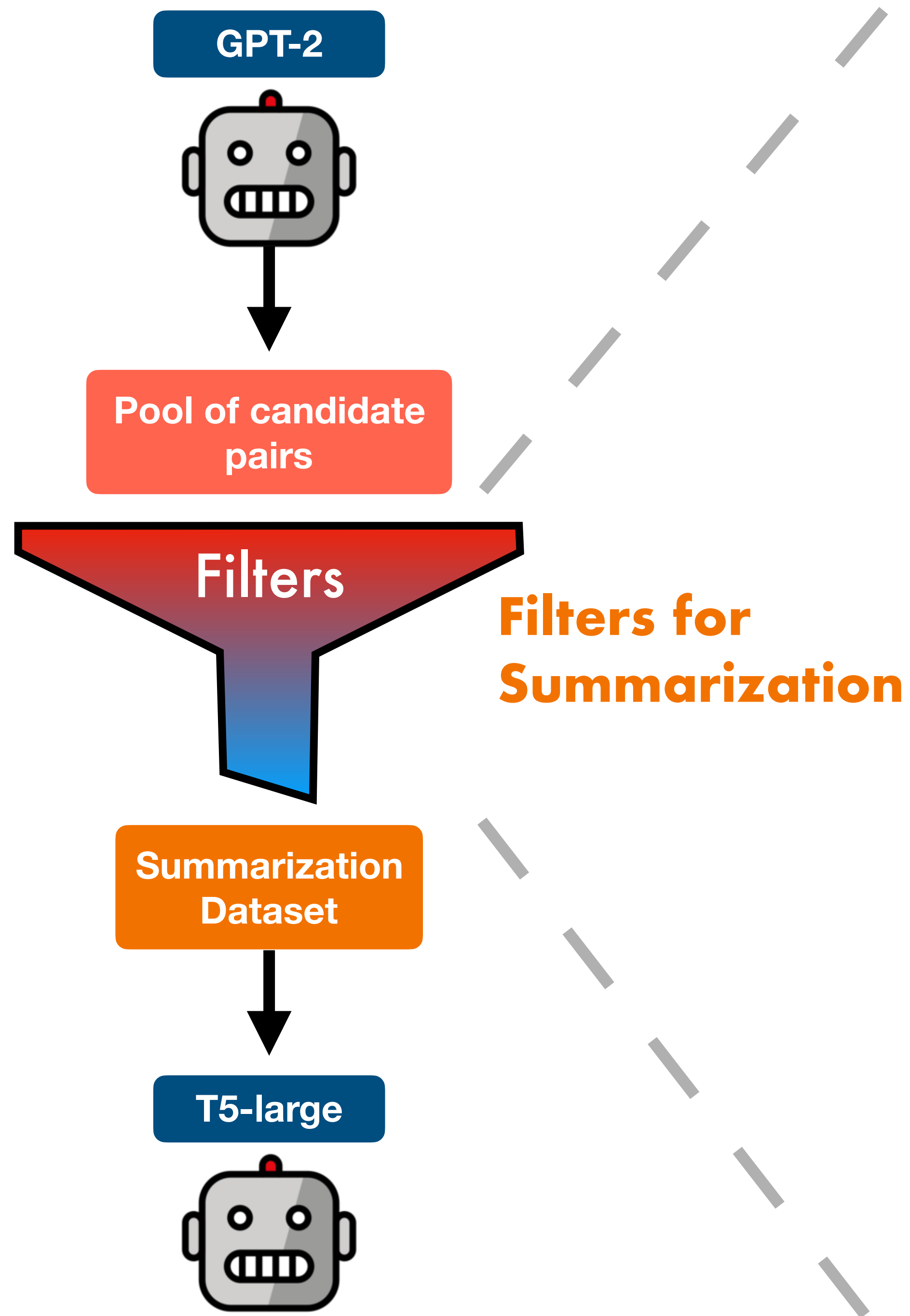
ATOMIC-10x: a machine-authored KB that wins, for the first time, over a human-authored KB in all criteria: scale, accuracy, and diversity.

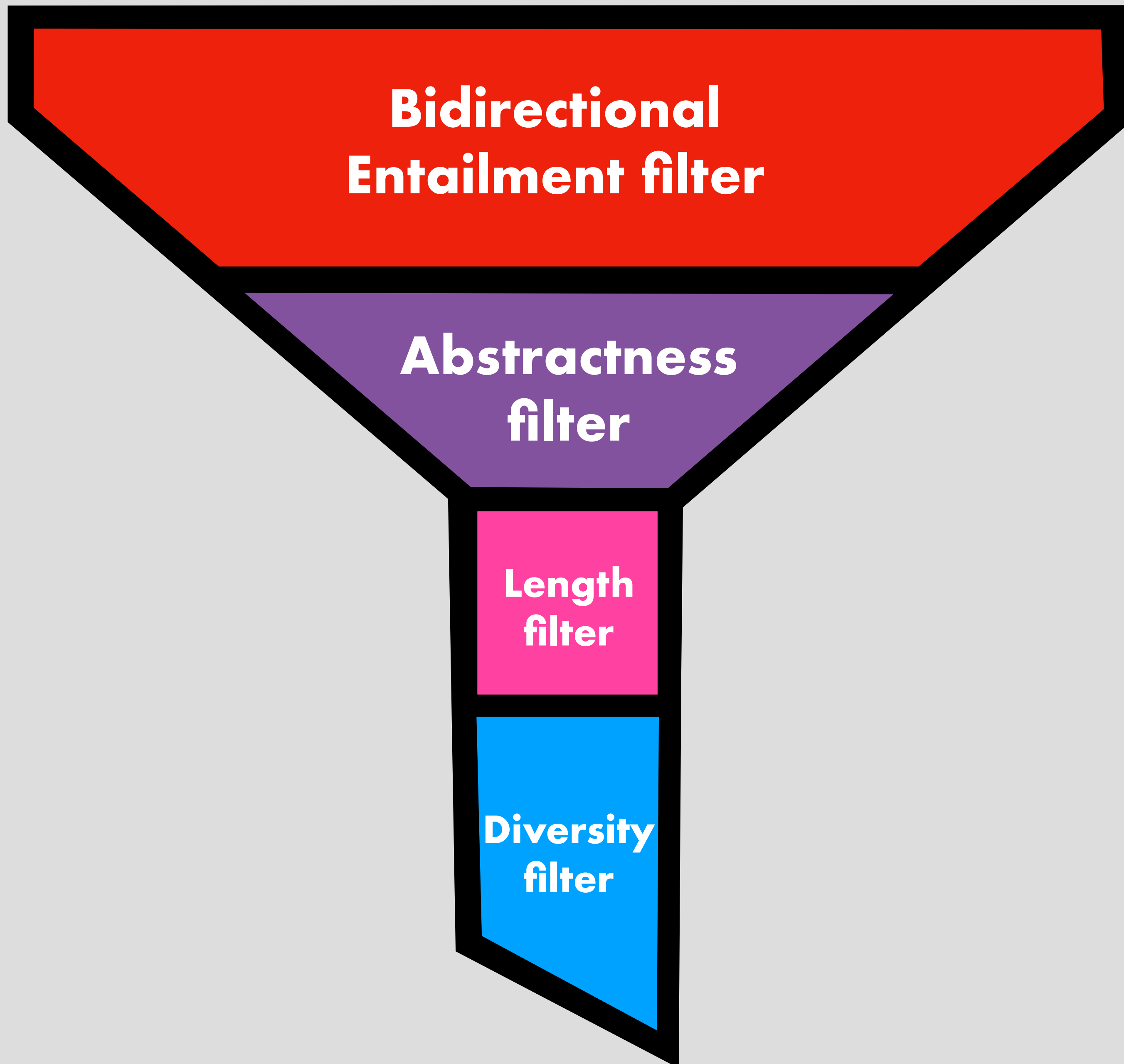


Yeah but can we get anywhere without GPT-3?

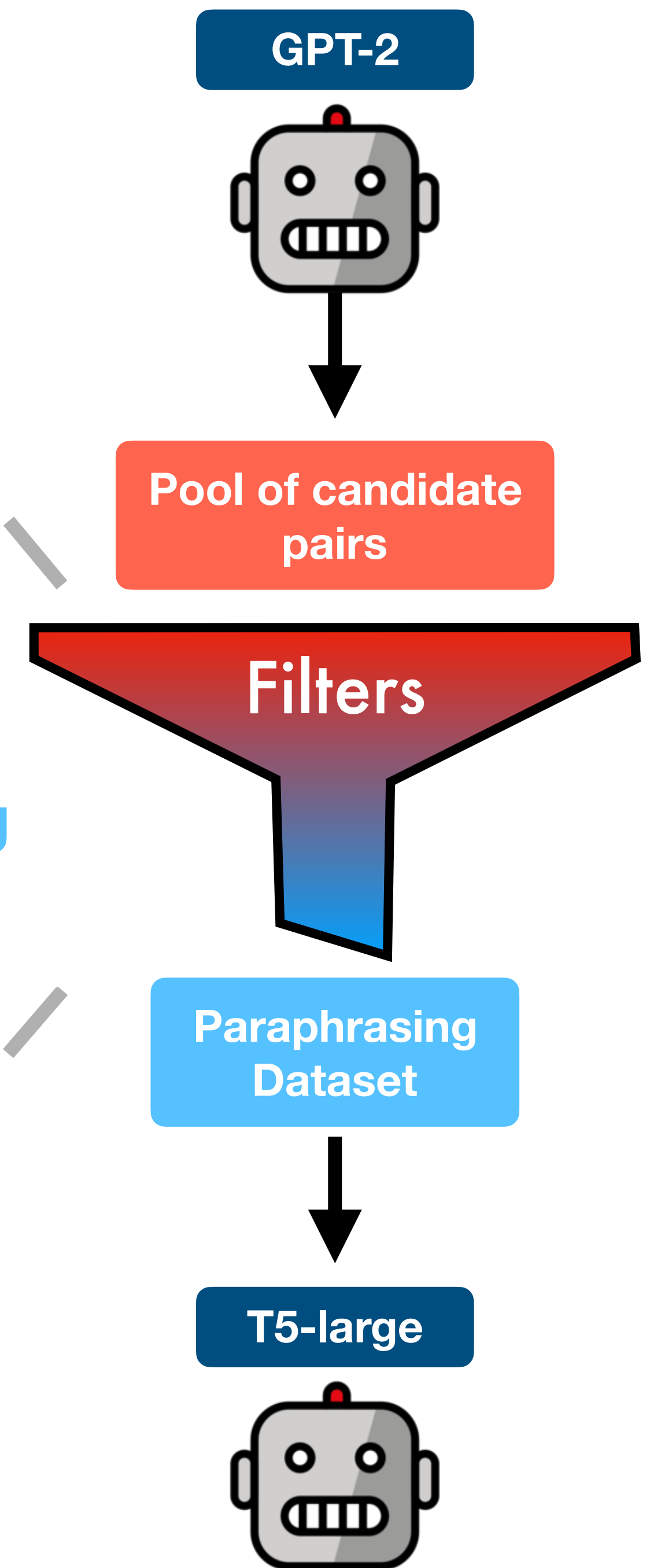


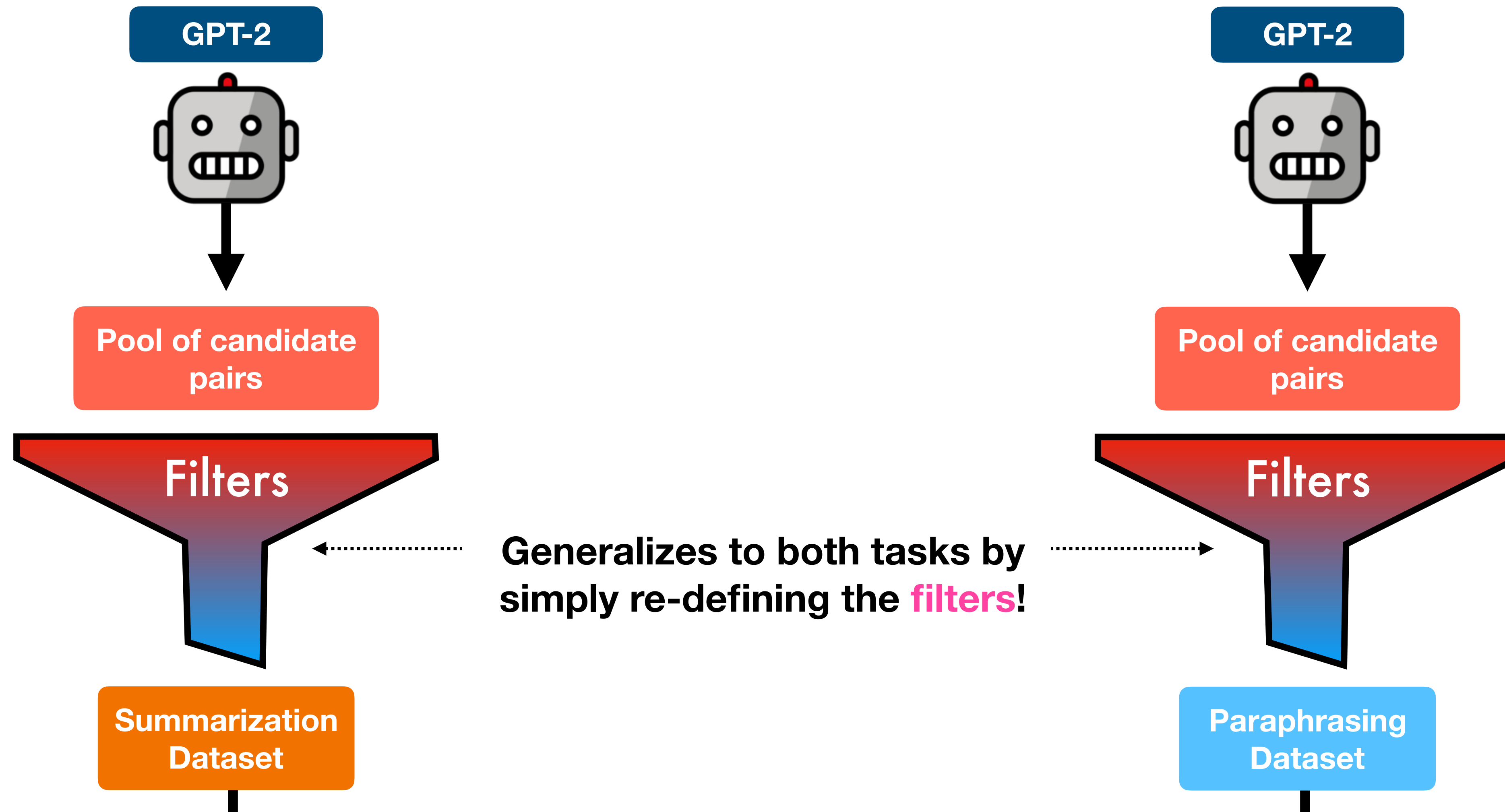




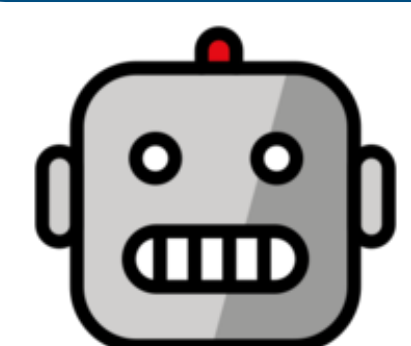


Filters for Paraphrasing

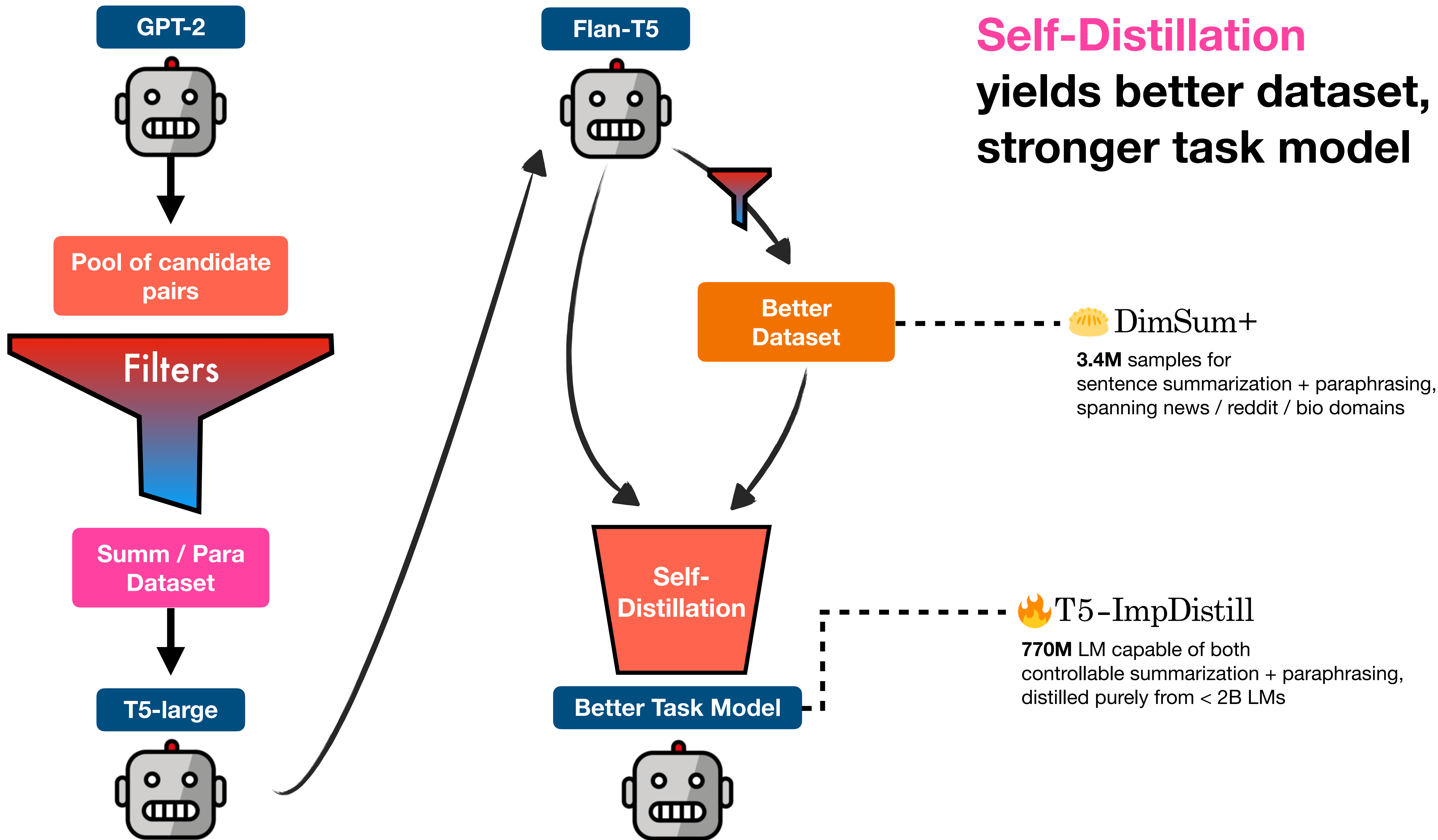




Flan-T5

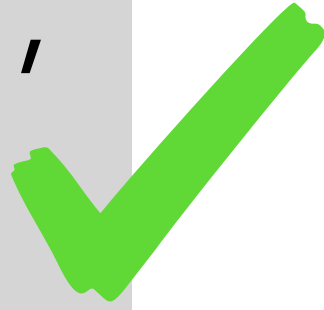


Train a single model capable of both tasks!



"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. 

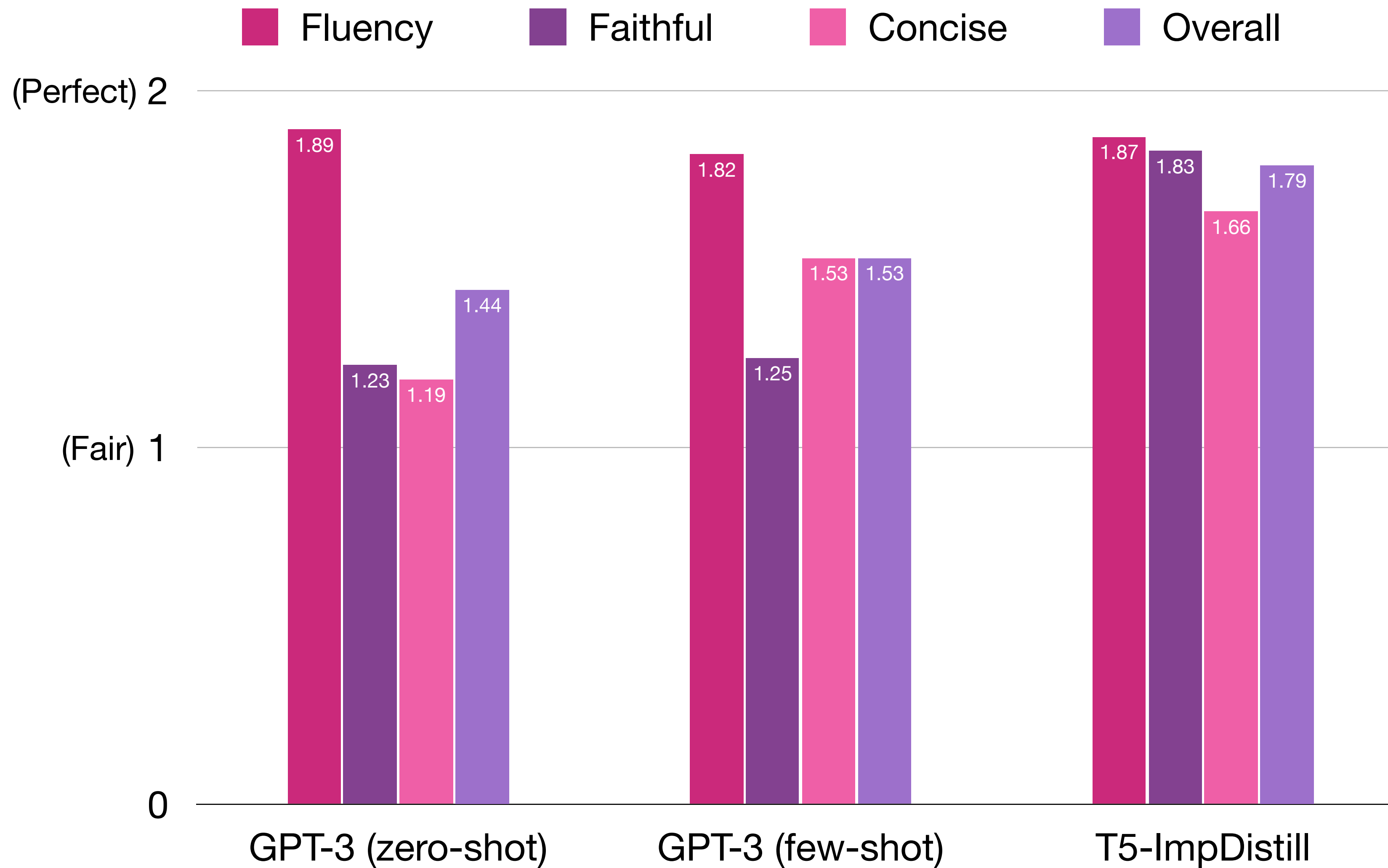
GPT-3 *text-davinci-003, zero-shot*

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.



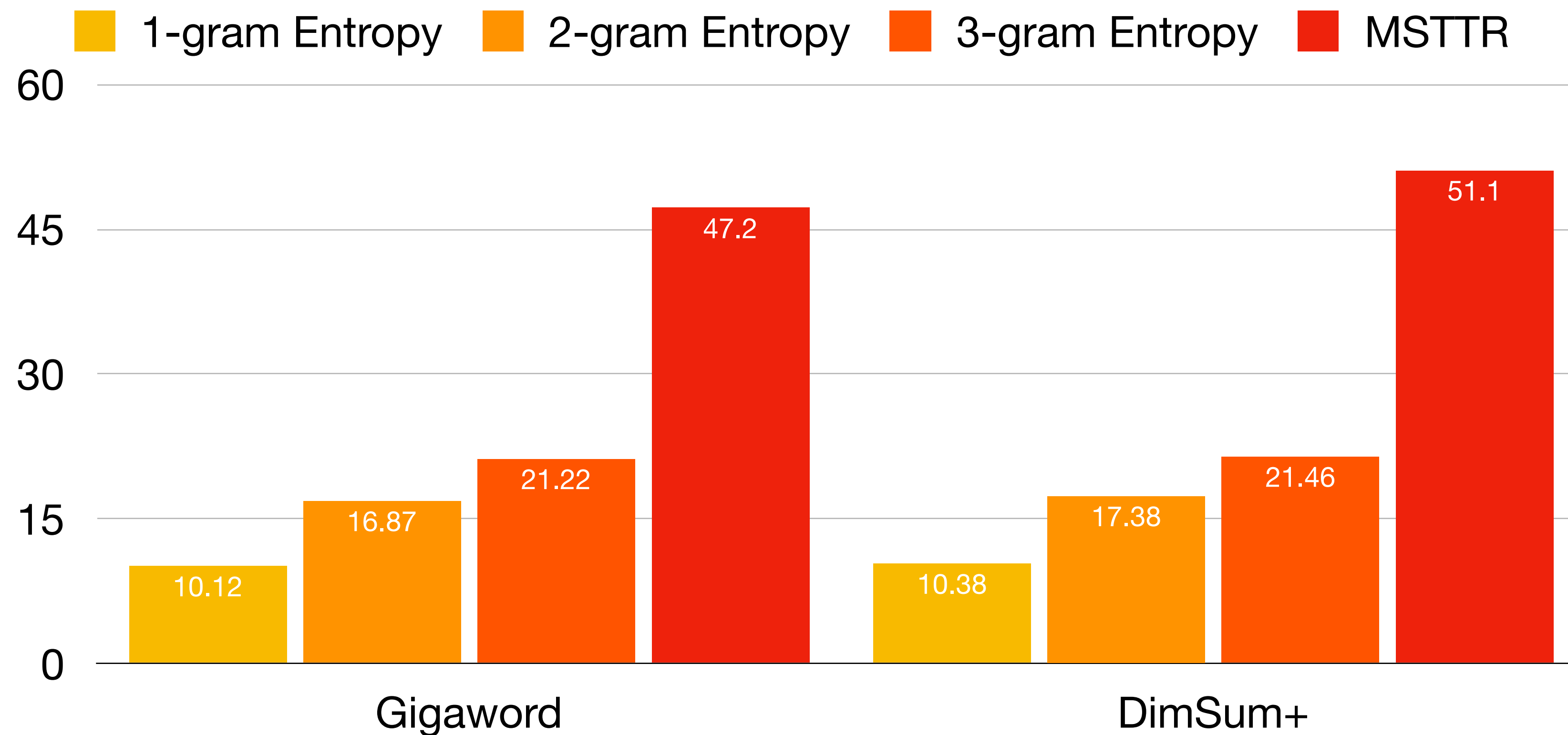
hallucinating unsupported content

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



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

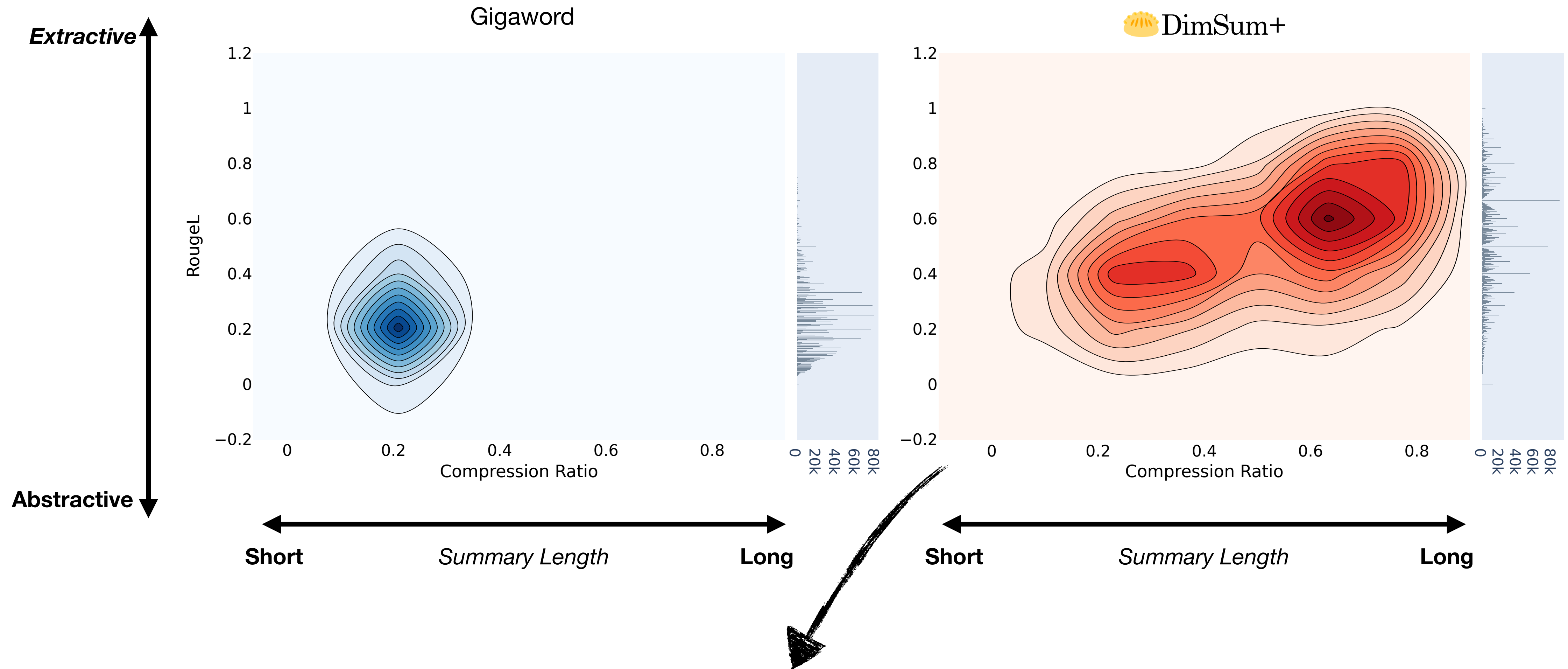
(Rush et al. 2015)



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!**

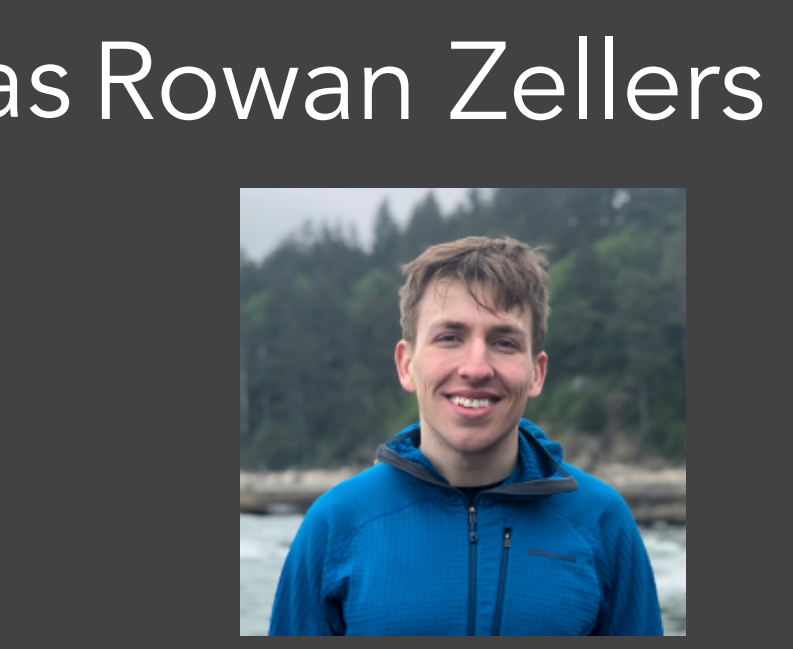
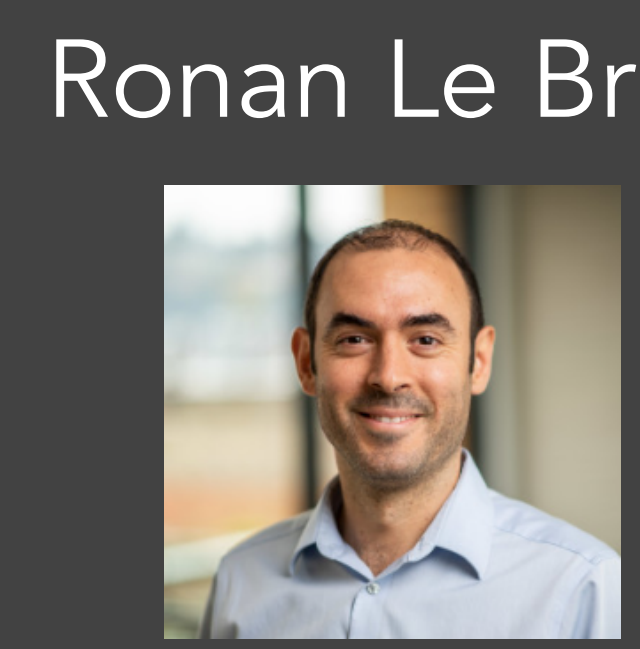
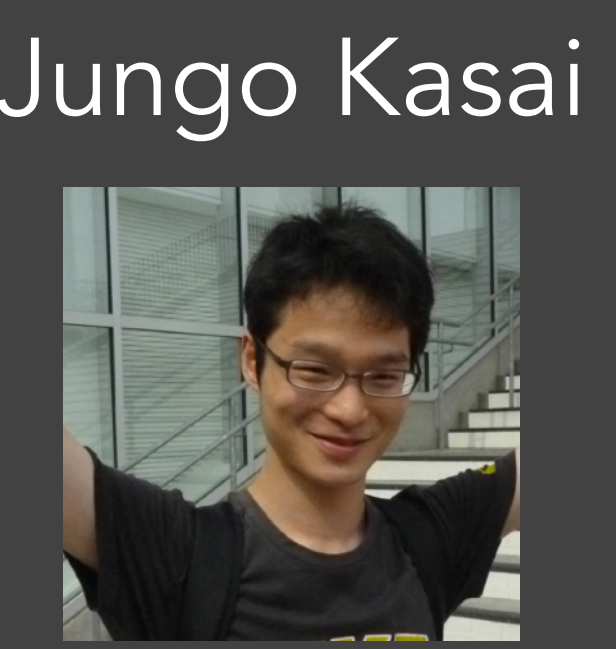
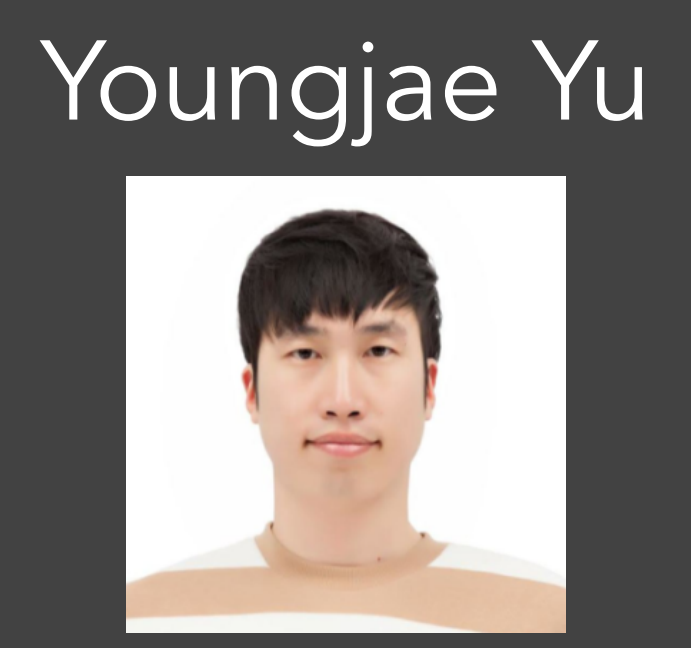
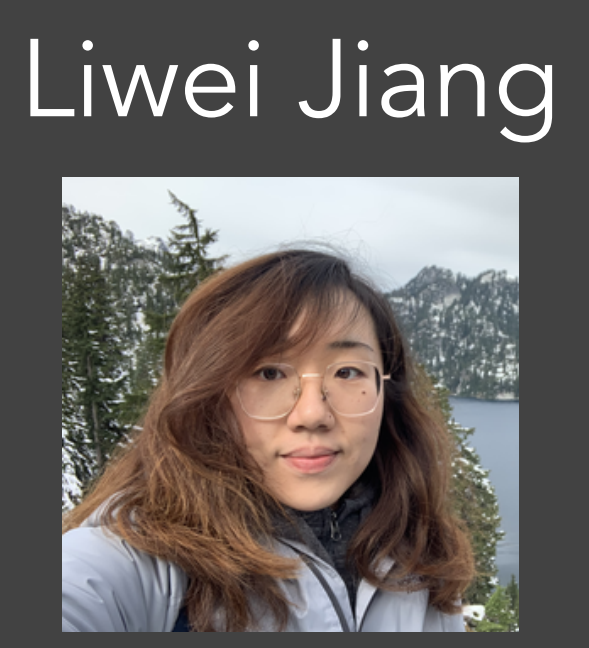
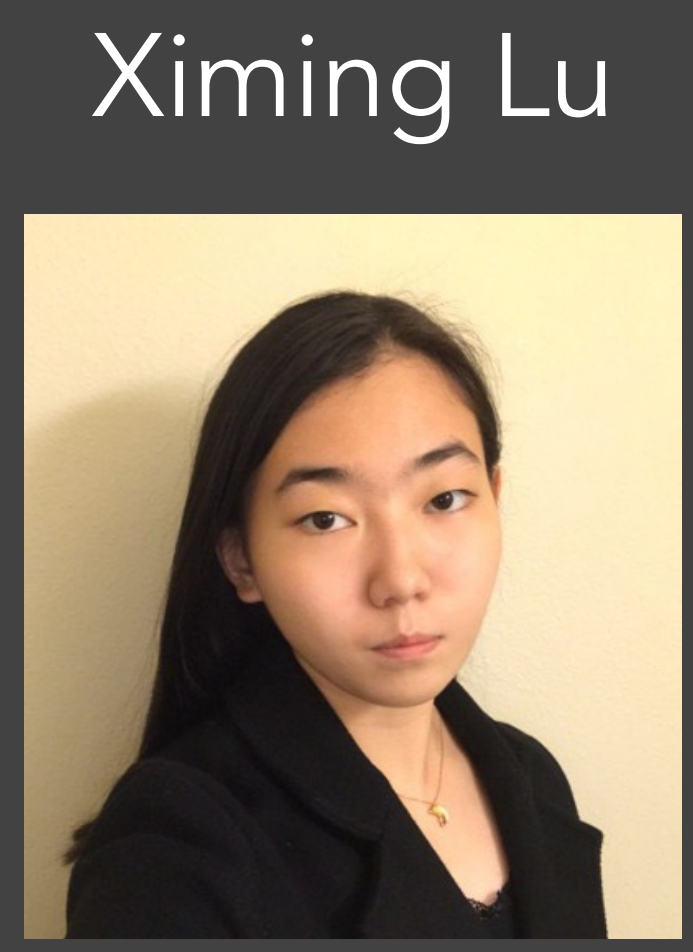


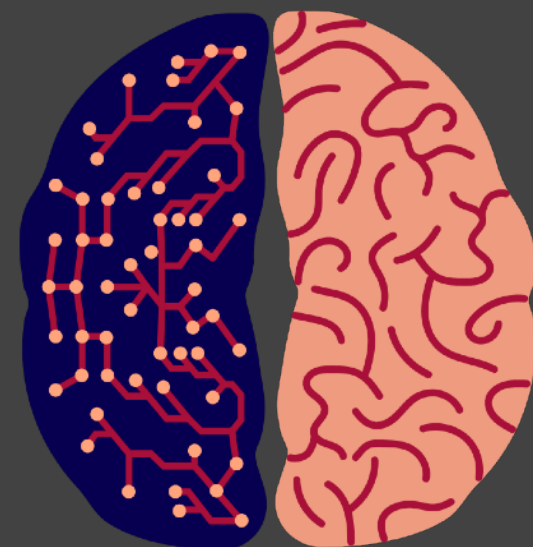
— 🏆 Best Method Paper Award at NAACL 2022 🏆 —



NEUROLOGIC A[★]ESQUE

Constrained Text Generation with Lookahead Heuristic





NEUROLOGIC DECODING

(Un)supervised Neural Text Generation with Predicate Logic Constraints

—NAACL 2021—

Ximing Lu



Peter
West



Rowan
Zellers



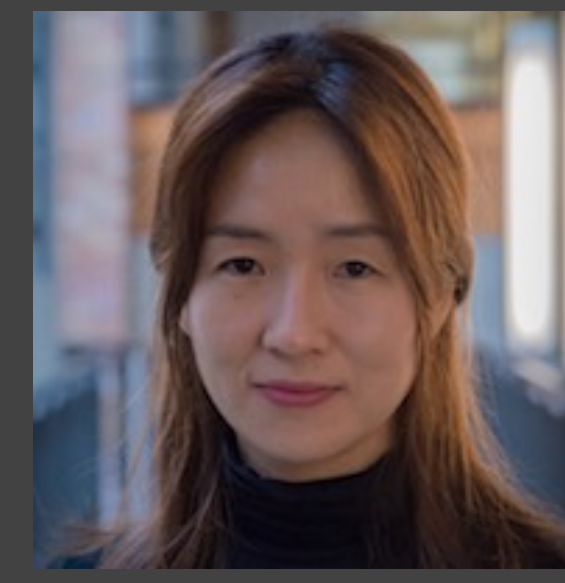
Ronan
LeBras



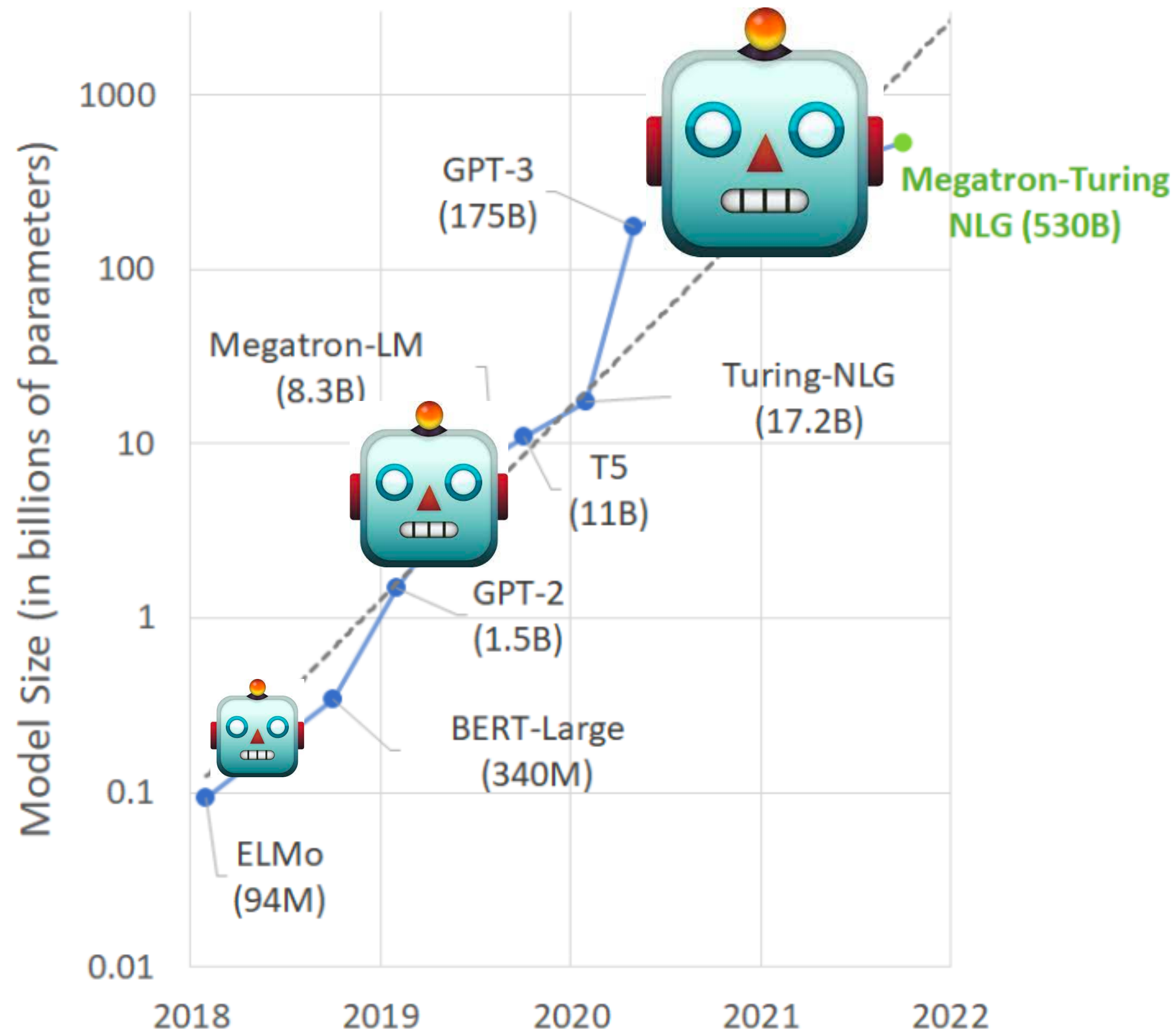
Chandra
Bhagavatula



Yejin
Choi



Neural Language Models



Program Synthesis

Your AI pair programmer

prompt: Write a python function that checks if a number is prime, one less than twice it, and satisfies these tests:

```

assert check(70) == False
assert check(23) == True
assert check(73) == True
    
```

model:

```

def check(n):
    if n == 2*int(str(n)[::-1]):
        return True
    else:
        return False
    
```

GitHub Copilot

Long-form QA

How long has AI been growing exponentially in the last 50 years?

175B best-of-64

many explanations for the exponential growth in technology in the last century. One explanation is that the pace of technological progress has accelerated exponentially over time because of a common force pushing it forward^[3]. Another explanation is that each new generation of AI models 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.

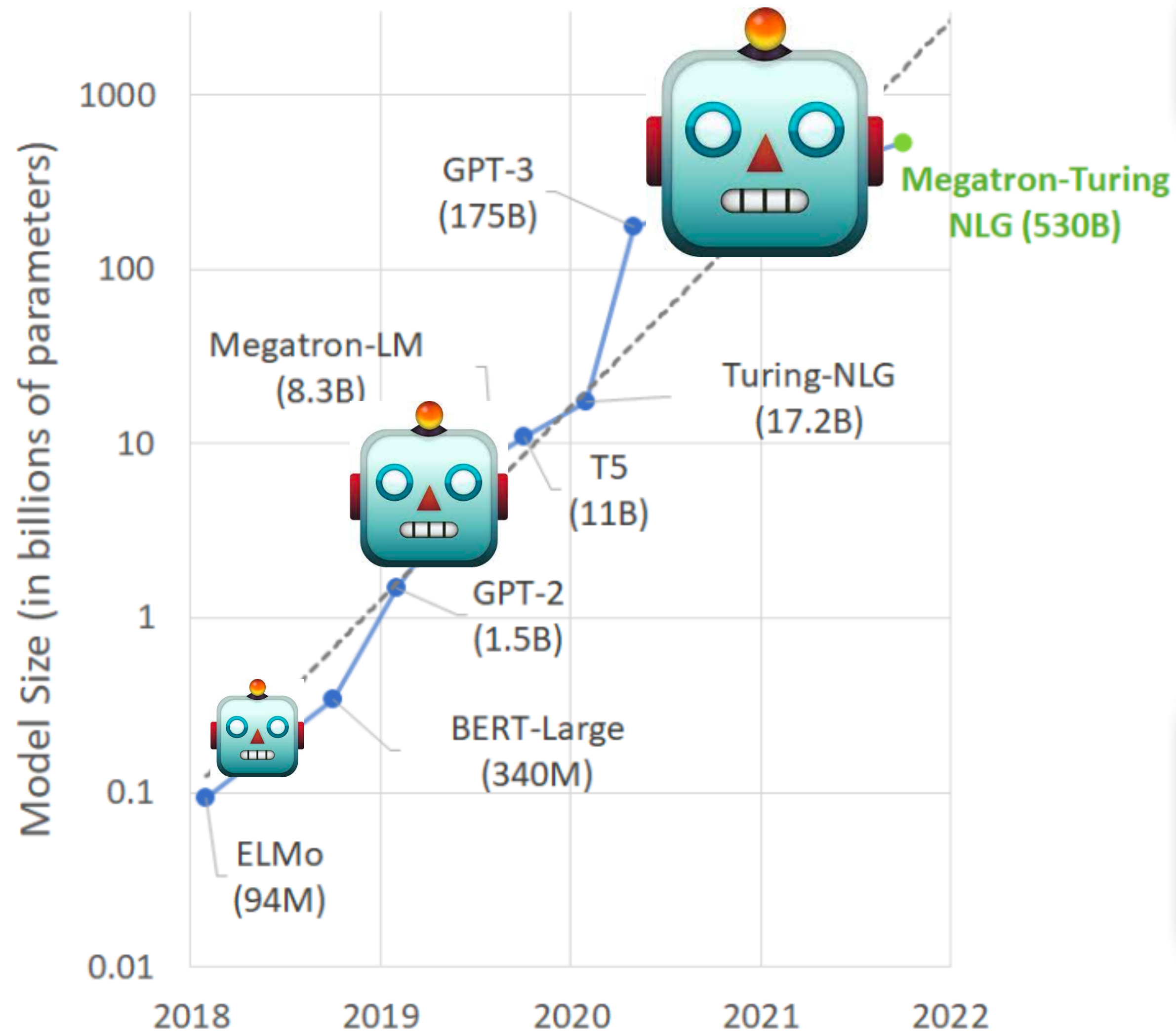
Machine Translation

Google Translate interface showing a text input field, language selection (English to Spanish), and a translation output area.

Dialogue

Dialogue interface showing a chat history with a user asking "What's a good topic for a new blog?" and a model responding "There are so many! How about something like a new food item that you just tried."

Neural Language Models



COMMONGEN

(Liu et al 2020)

What is the mass of Jupiter?

Language Model
(GPT3)

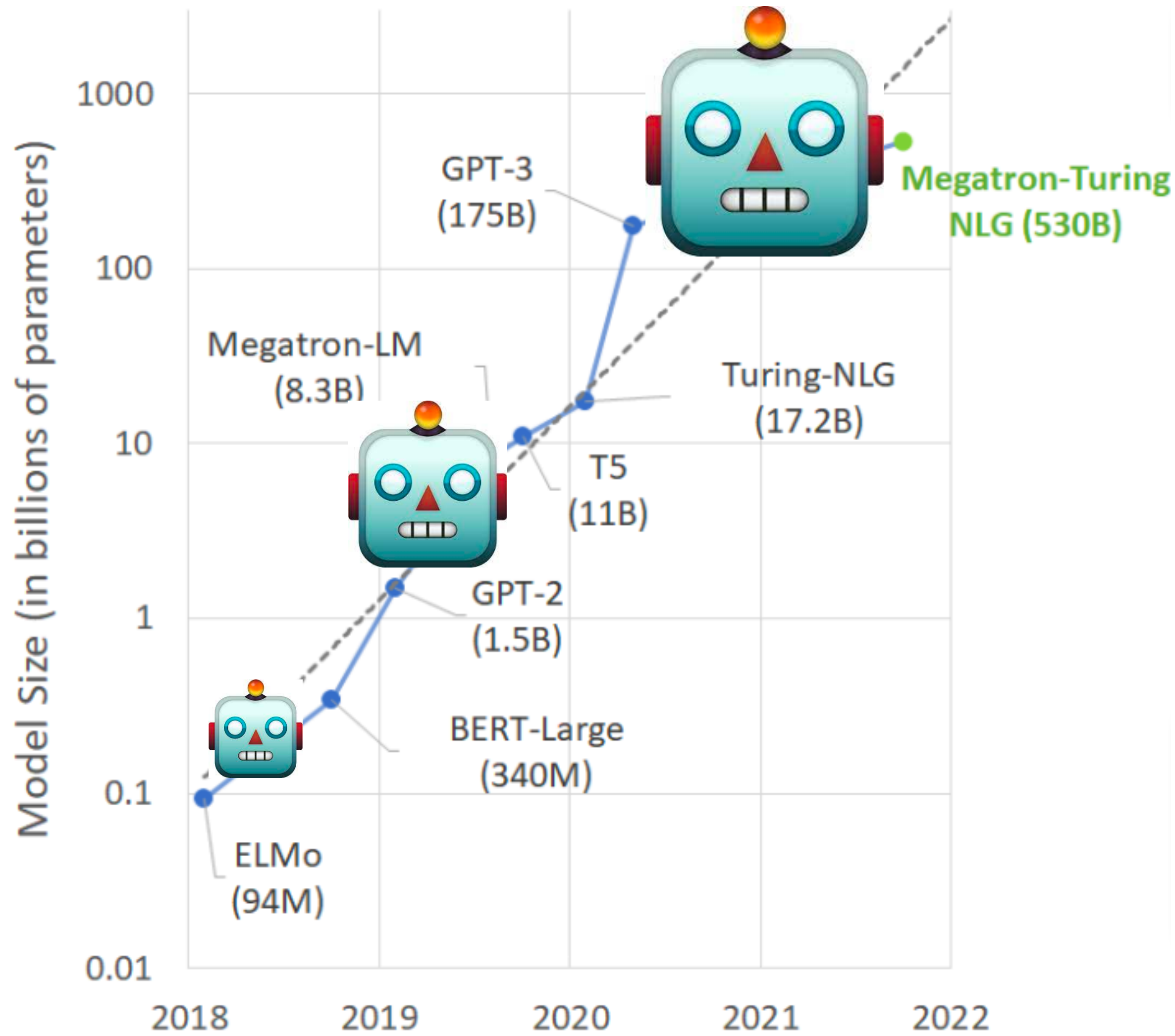
Generate a question containing all of the given words.

Words: Jupiter, Mercury, Venus, mass



missing keywords

Neural Language Models



Search Algorithms in **Classical AI**

A* Search

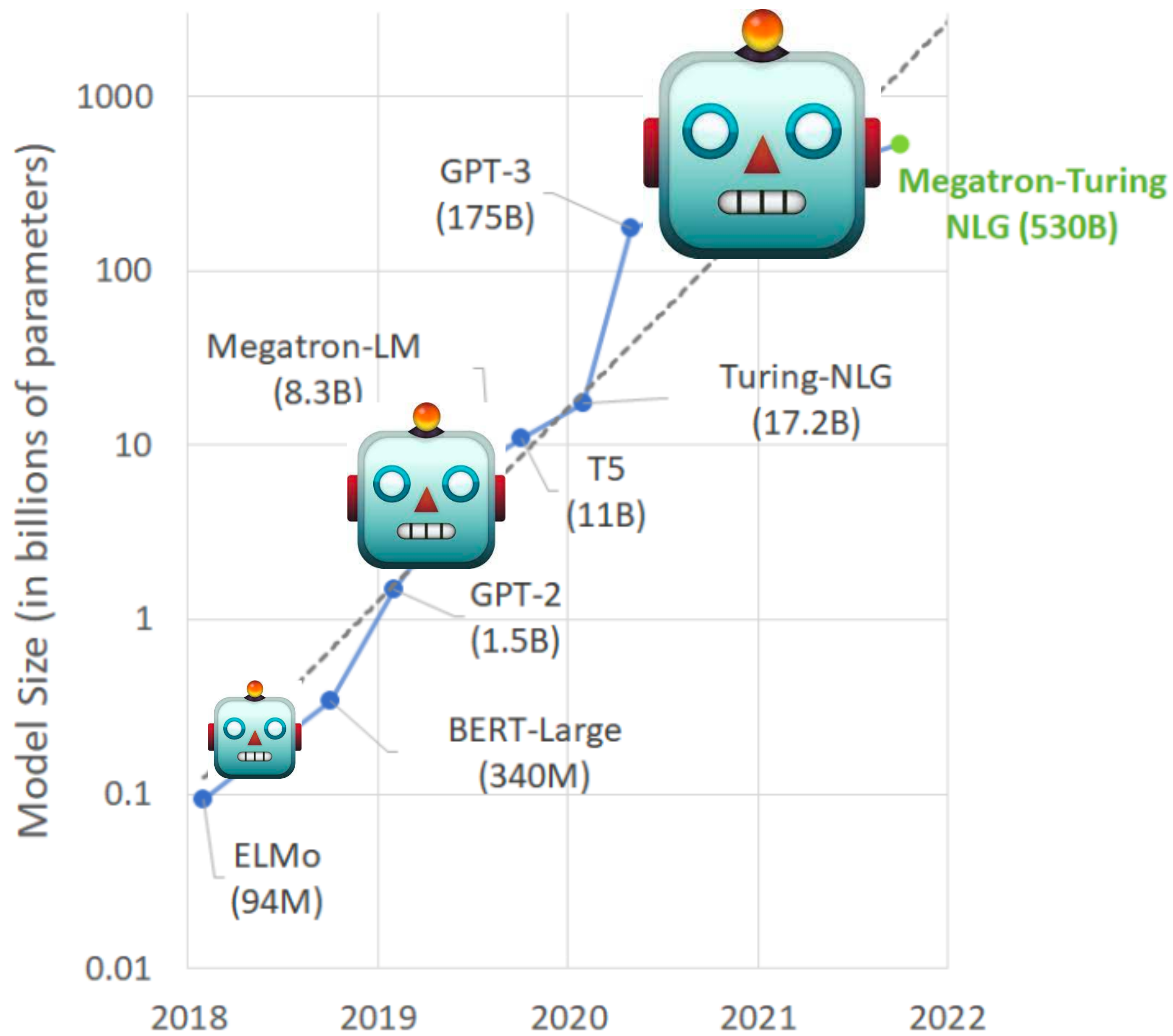
MinMax Search

Monte Carlo Tree Search

Best-first Search

Dijkstra's

Neural Language Models



Search Algorithms in **Classical AI**

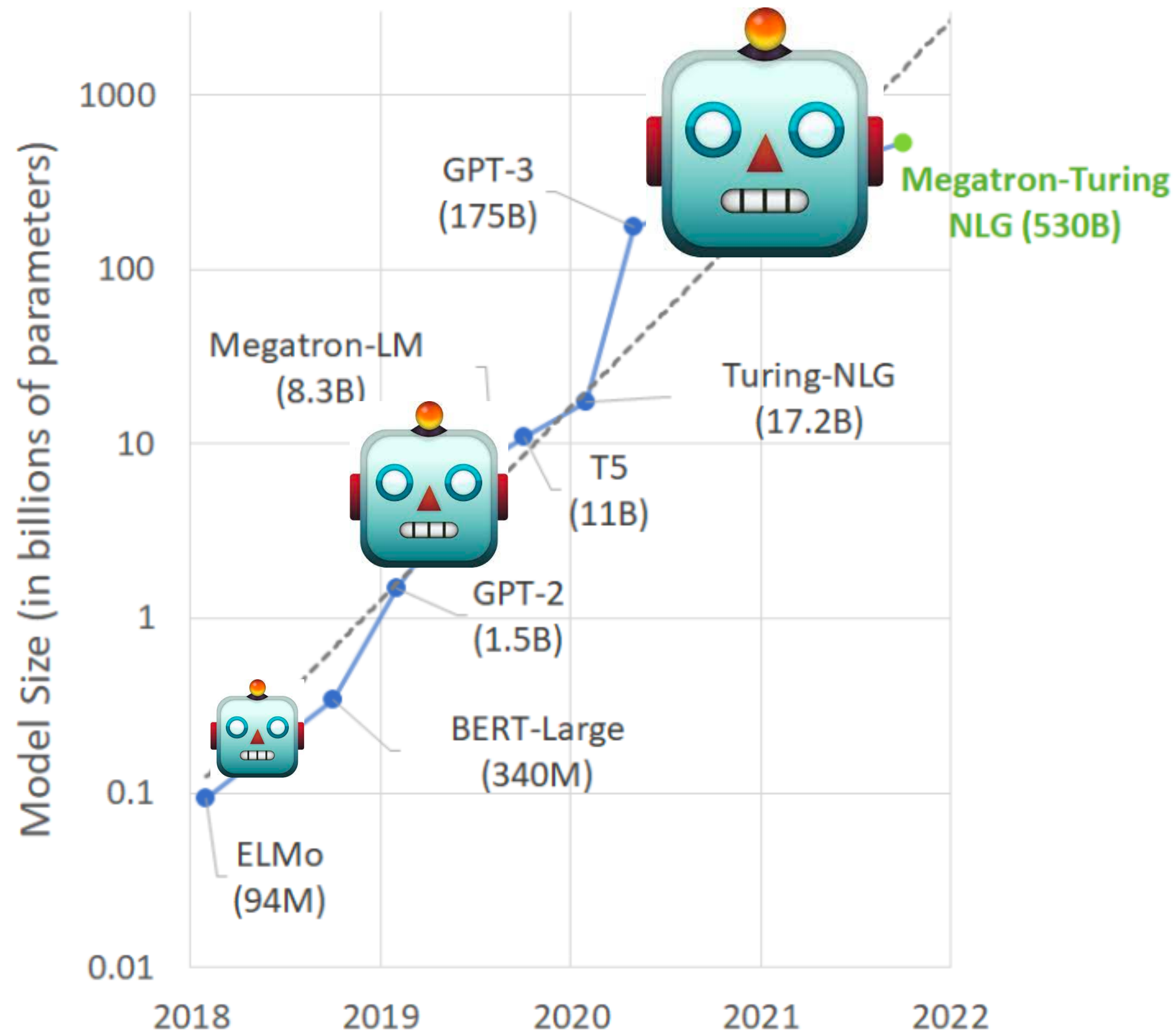
A* Search

MinMax Search

Monte Carlo Tree Search

AlphaGO

Neural Language Models



Search Algorithms in **Classical AI**

A* Search

MinMax Search

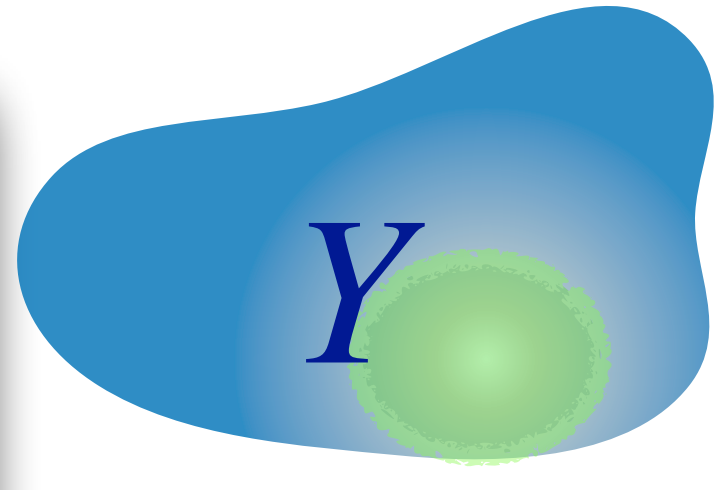
Monte Carlo Tree Search

Best-first Search

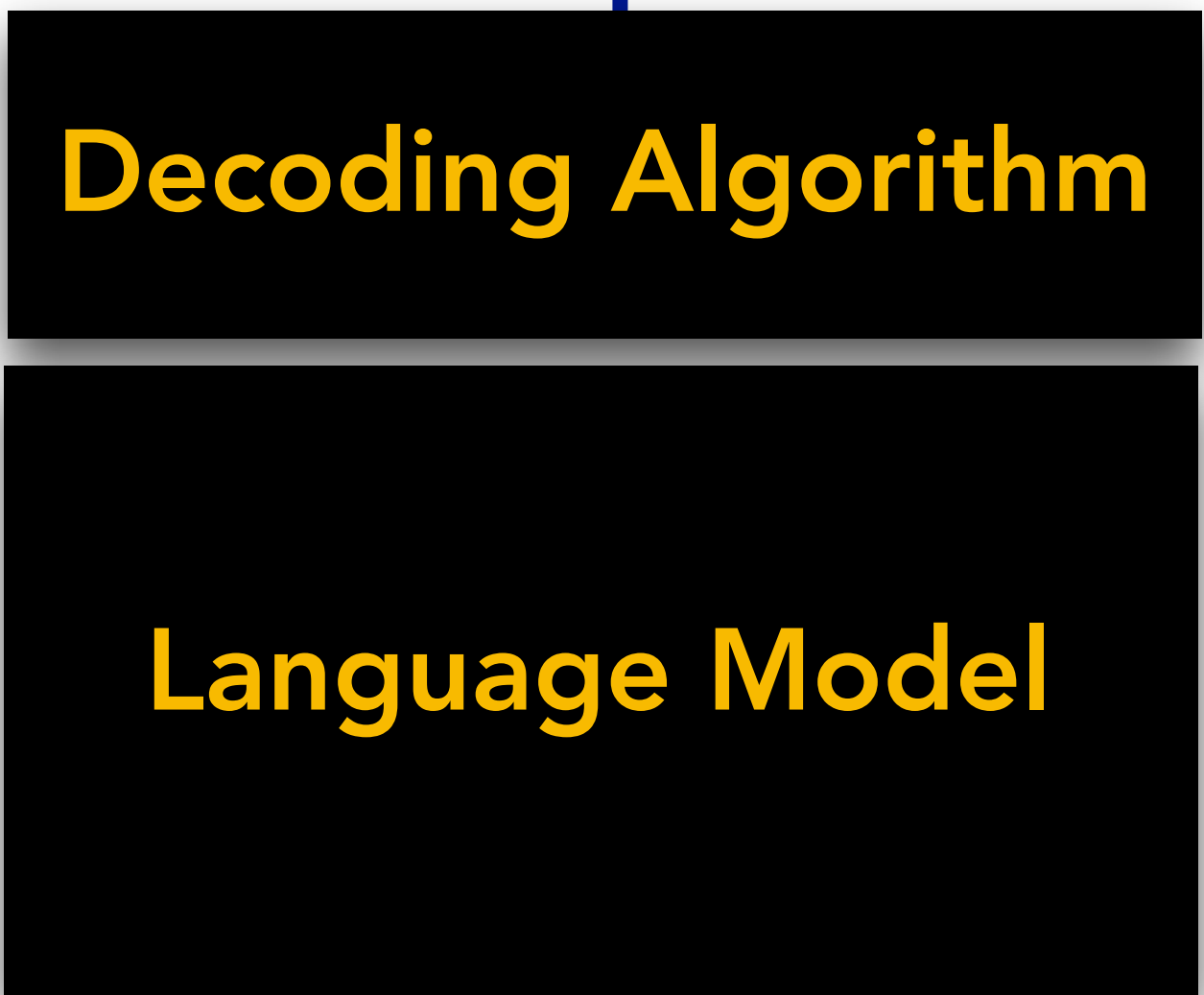
Dijkstra's

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

Logical Constraints
 (Jupiter) \wedge (Mercury) \wedge
 (Venus) \wedge (mass \vee masses)



C



What is the mass of Jupiter?

GPT3

Generate a question containing all of the given words.
 Words: Jupiter, Mercury, Venus, mass
 ✓ ✗ ✗ ✓

X

Table to Text

X

type	hotel
count	182
dogs allowed	don't care

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

Image Captioning

X

Y
 A giraffe standing in a field with a zebra.

Machine Translation

X
 Silent night: Tips to fight sleep disorders.

Y
 Erholung Nacht: Tipps gegen Schlafstörungen.

NeuroLogic Decoding in a Nut Shell

Constraints $\underbrace{\diamond_1(\text{cowboy}) \wedge \diamond_2(\text{dog})}_{\diamond_1} \wedge \underbrace{(\diamond_3(\text{play music}) \vee \diamond_4(\text{plays music})) \wedge (\diamond_5(\text{catch}) \vee \diamond_6(\text{catches}))}_{\diamond_2}$

search tree	likelihood	clauses	score	select	notation
	0.18	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	$0.18 + 0.1 * 0 = \underline{0.18}$	✓	<p>$\square\square\square\square$ denotes the state for $\diamond_1, \diamond_2, \diamond_3, \diamond_4$ separately, \square indicates \diamond_i is irreversibly stratified, \square otherwise.</p> <p>Pruning step: \circ denotes failure in top-α filtering in term of likelihood, \circ denotes failure in top-β filtering in term of number of satisfied clauses</p> <p>Grouping step: $\textcircled{1} \textcircled{2} \textcircled{3} \textcircled{4}$ denotes candidate groups based on the shared set of irreversibly satisfied clauses</p> <p>Selecting step: $\underline{\quad}$ denotes the top-1 candidate within each group ranked by score function. Among these candidates, we select ✓ the top-k ones to fill in the next beam.</p> <p>$s = P_\theta(\mathbf{y}_t \mathbf{y}_{<t}) + \lambda \cdot \max_{\substack{D(\mathbf{a}_i, \mathbf{y}) \\ \in \text{state } S1}} \frac{ \hat{\mathbf{a}}_i }{ \mathbf{a}_i }$</p>
	0.12	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	$0.12 + 0.1 * 0 = \underline{0.12}$	✓	
	0.05	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>			
	0.20	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>			
	0.19	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>			
	0.16	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>			
	0.15	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	$0.15 + 0.1 * 0 = 0.15$		
	0.11	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	$0.11 + 0.1 * \frac{1}{2} = \underline{0.16}$	✓	
	0.09	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	$0.09 + 0.1 * 0 = \underline{0.09}$		

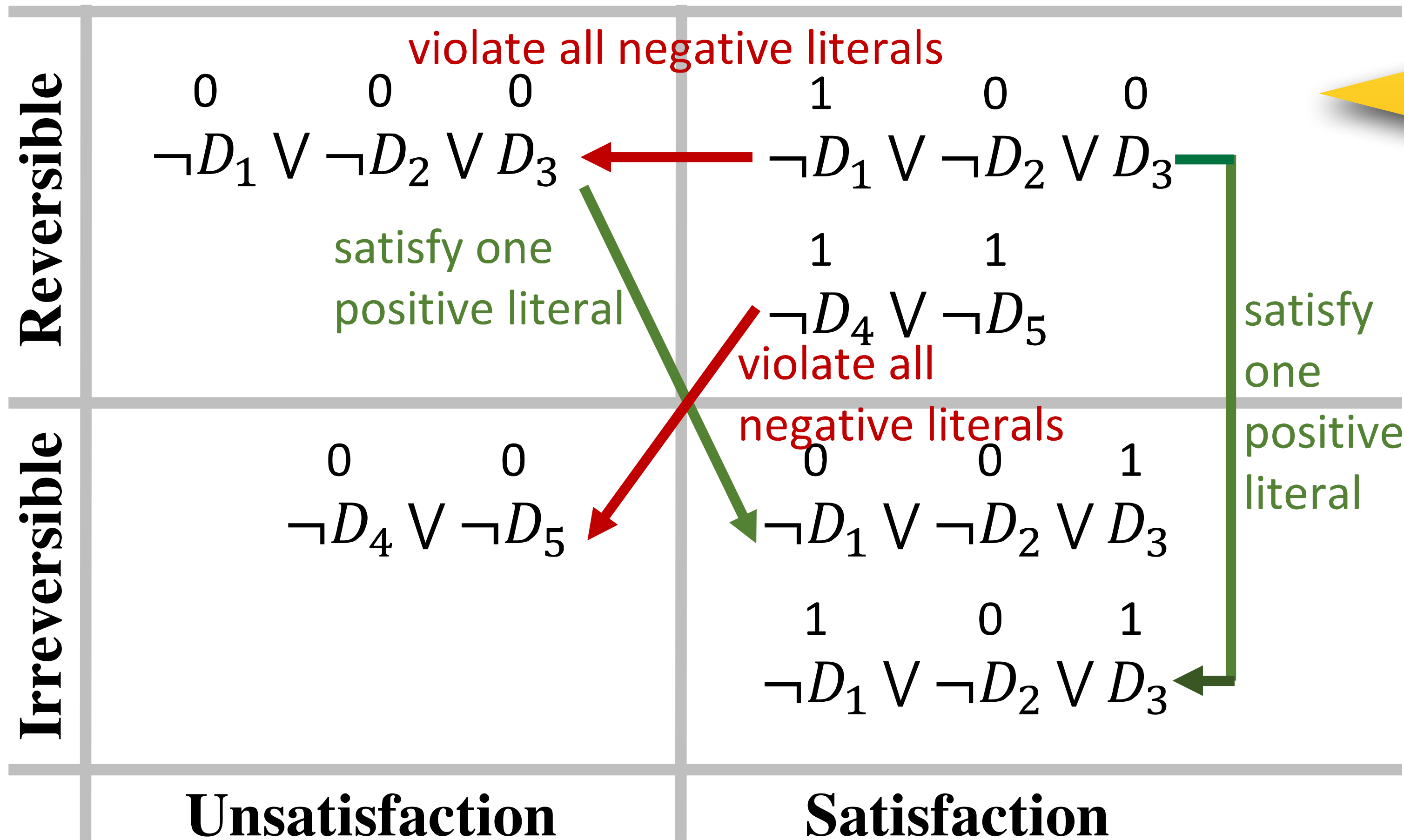
t=0

t=1

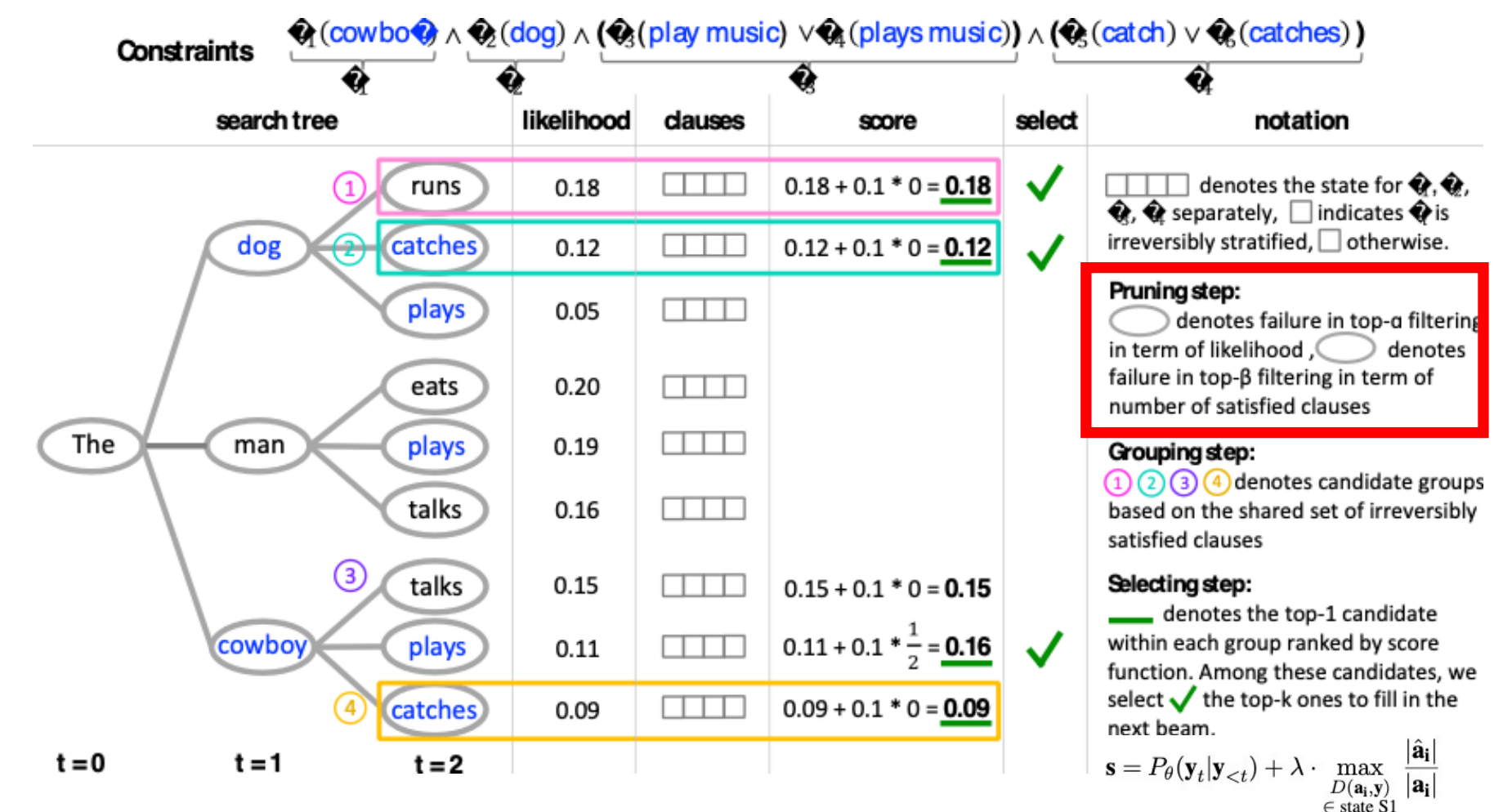
t=2

NeuroLogic Decoding in a Nut Shell

— it's a logic-guided search algorithm



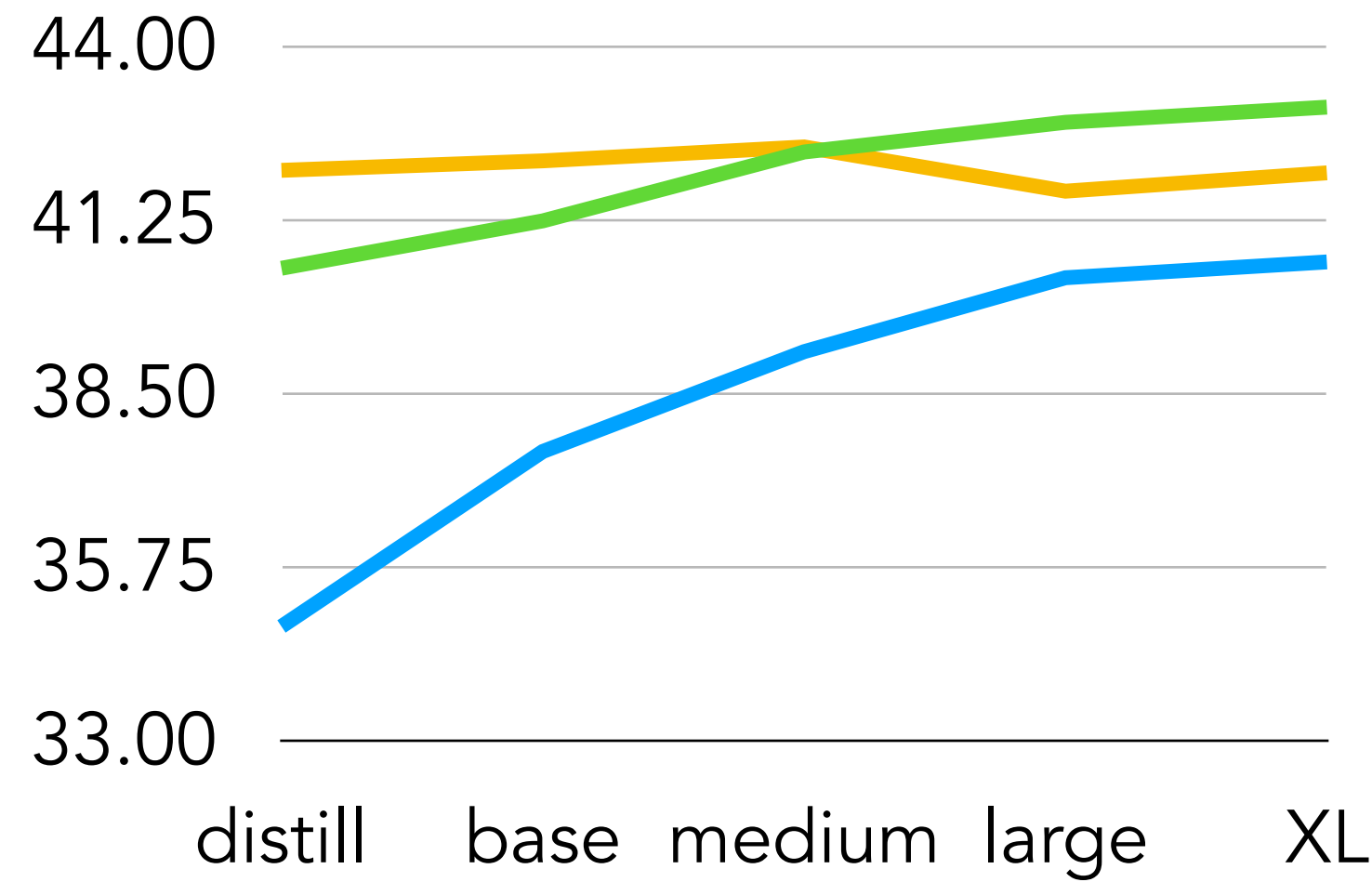
- four states of clause satisfaction:
- reversible satisfaction
 - irreversible satisfaction
 - reversible unsatisfaction
 - irreversible unsatisfaction



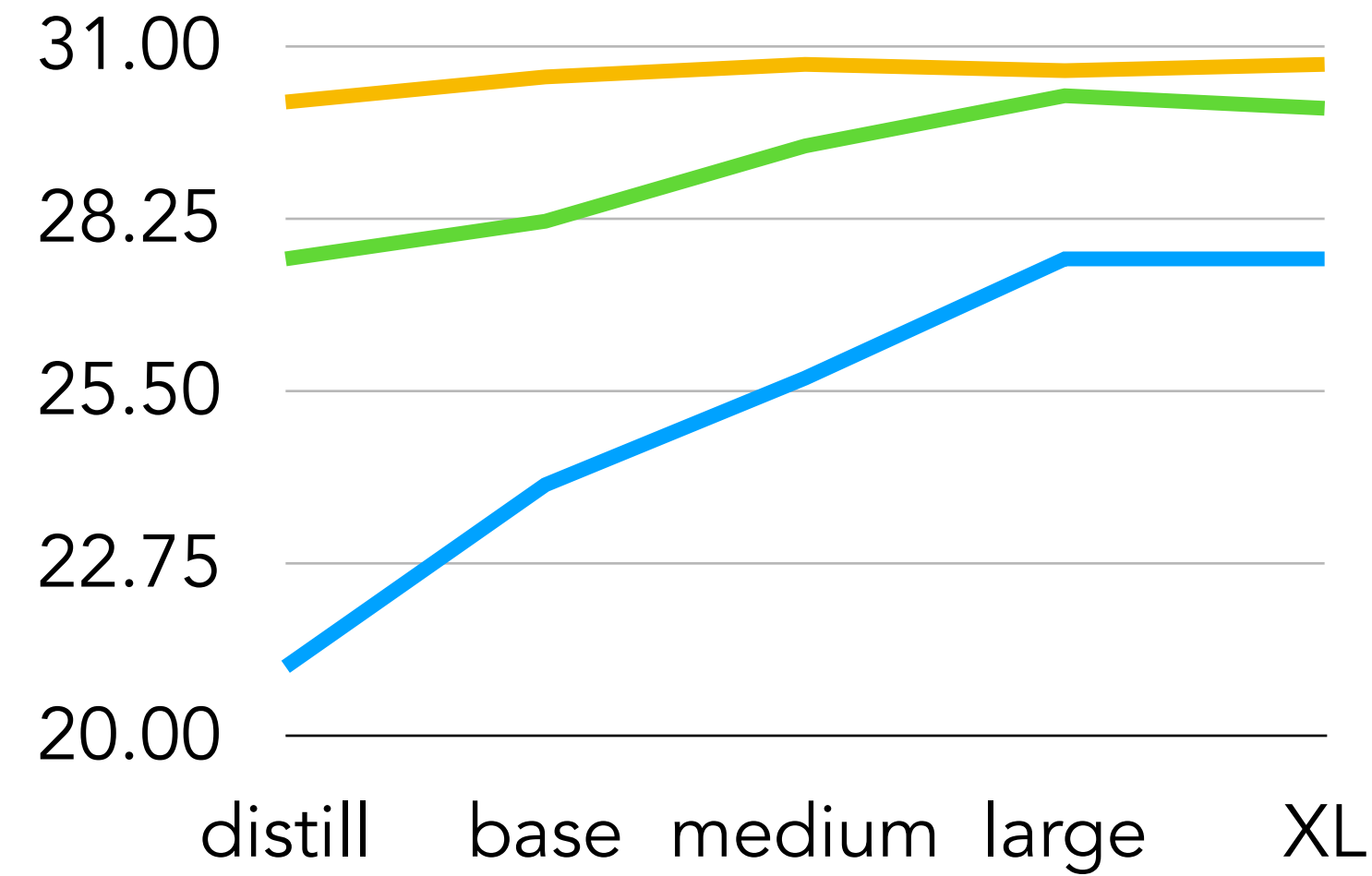
COMMONGEN (Zero-shot)

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

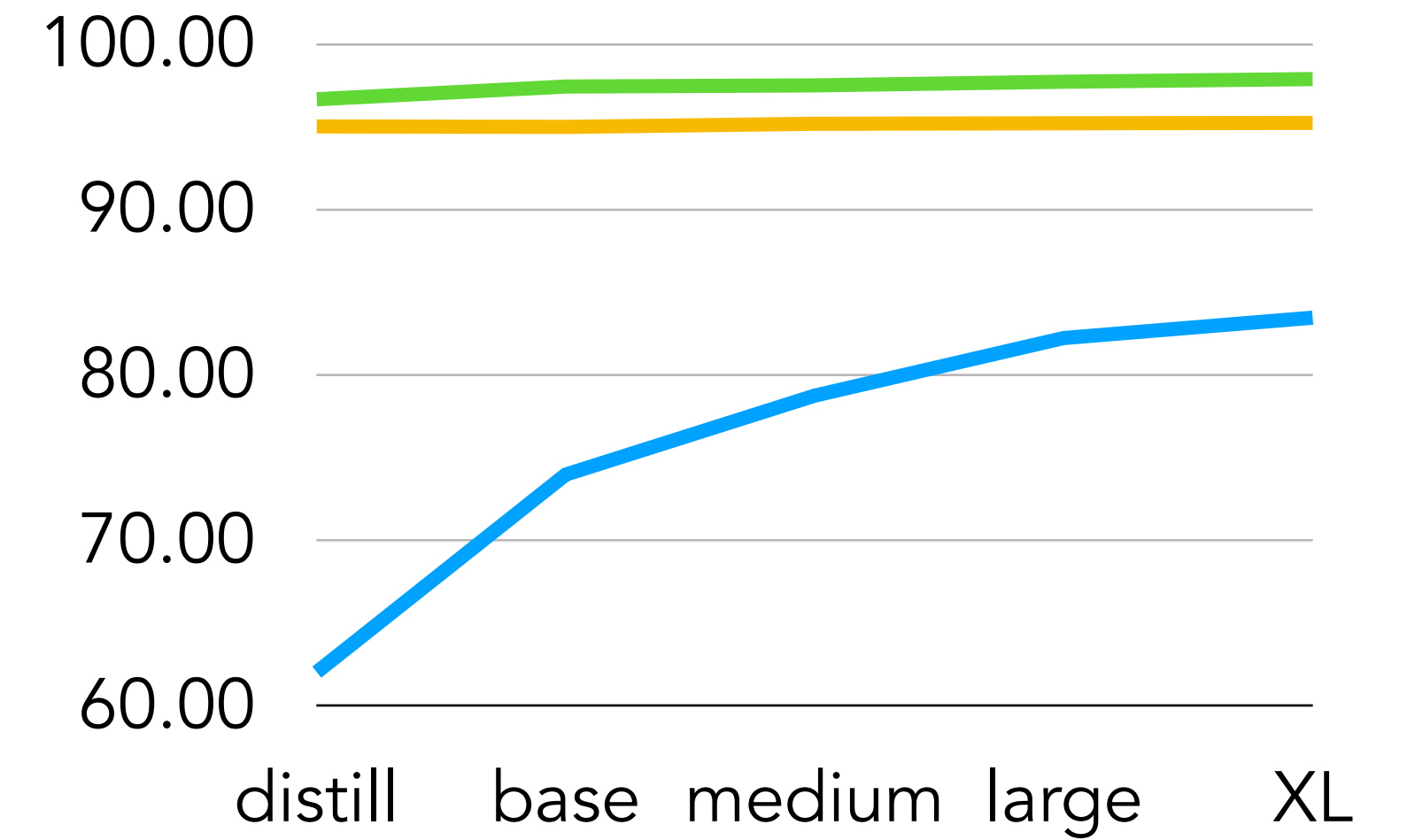
ROUGE-L



METEOR

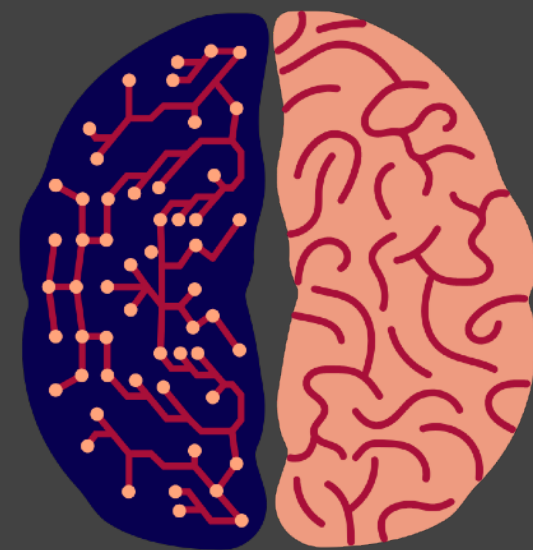


Coverage



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—

PAUL G.
ALLEN
SCHOOL

W

— 🏆 Best Method Paper Award at NAACL 2022 🏆 —

Ai2

NEUROLOGIC A[★]ESQUE

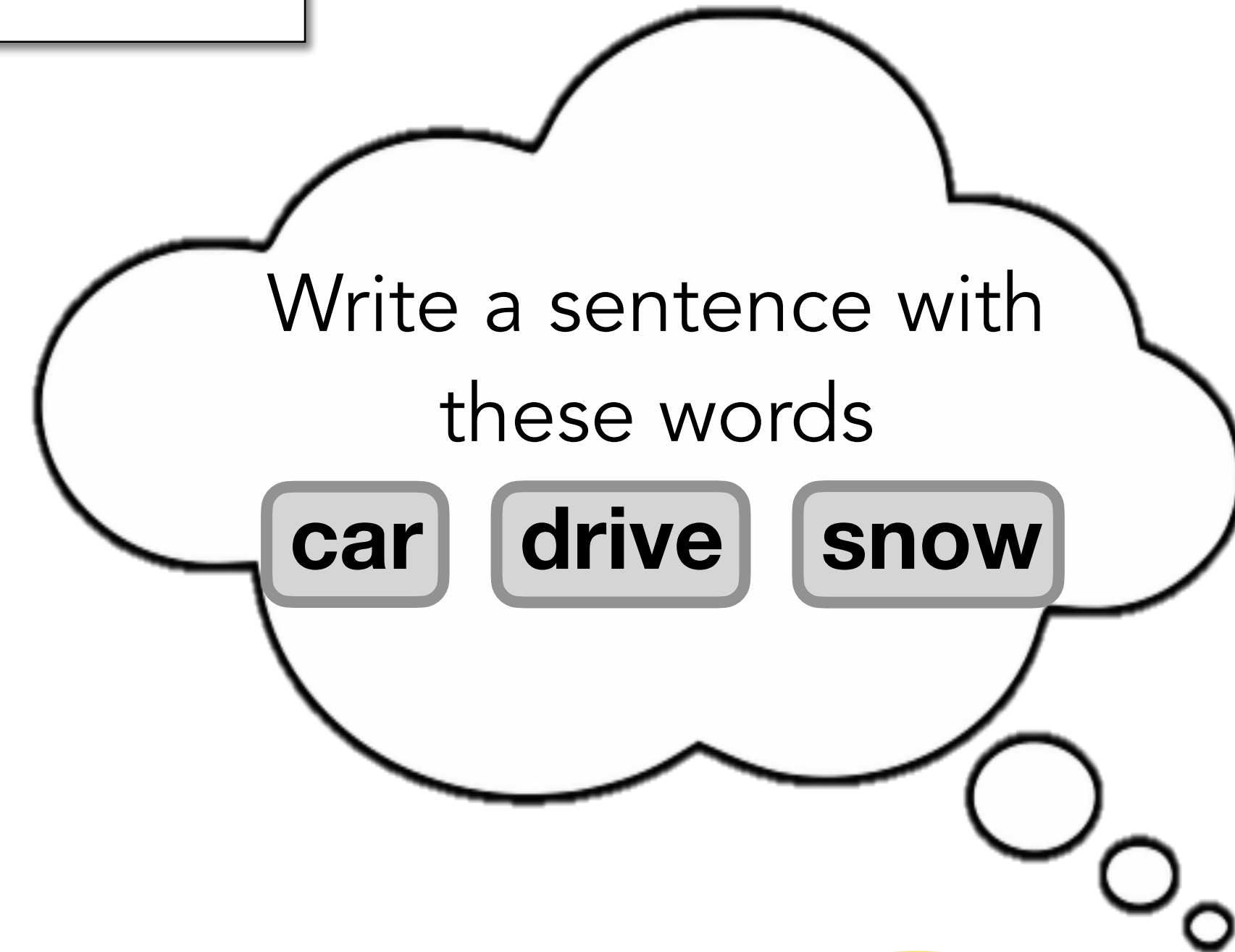
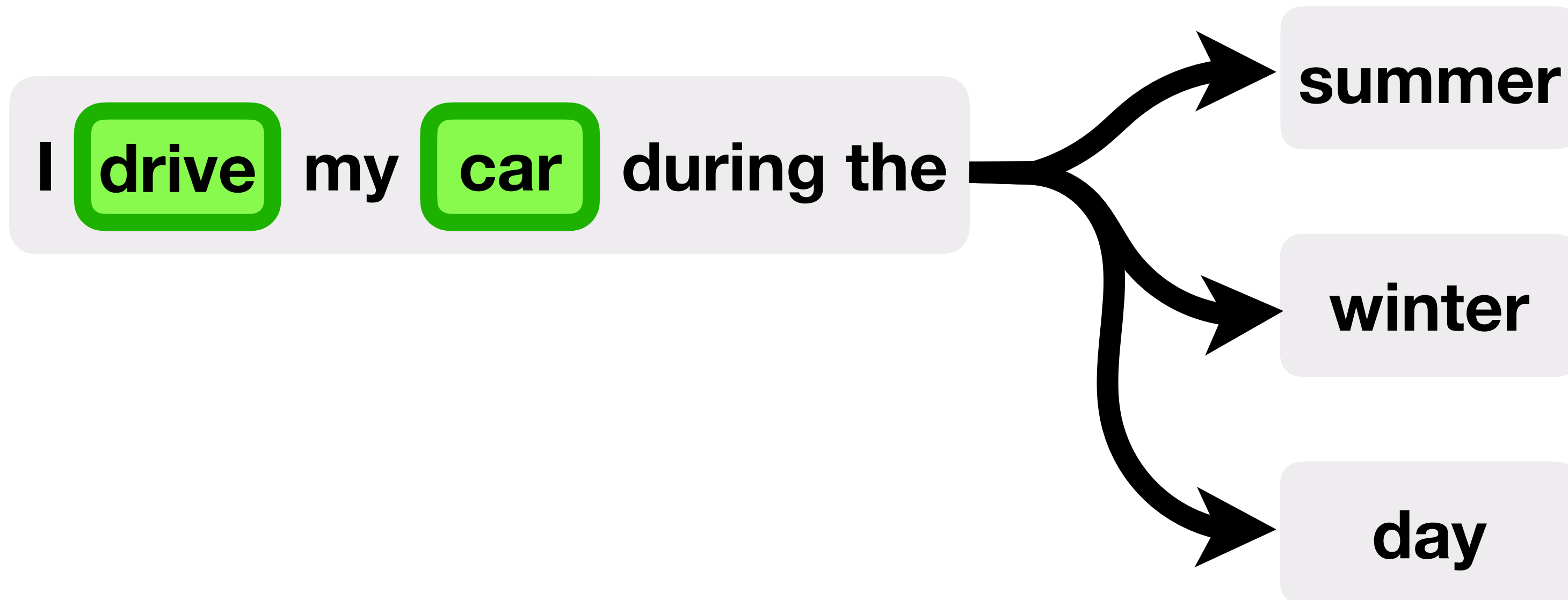
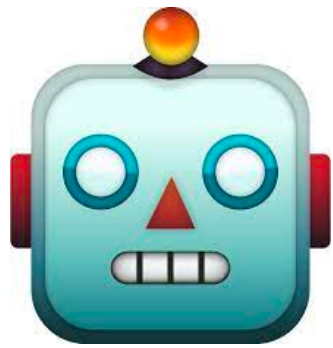
Constrained Text Generation with
Lookahead Heuristic

NeuroLogic Decoding

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

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

Off-the-Shelf GPT2

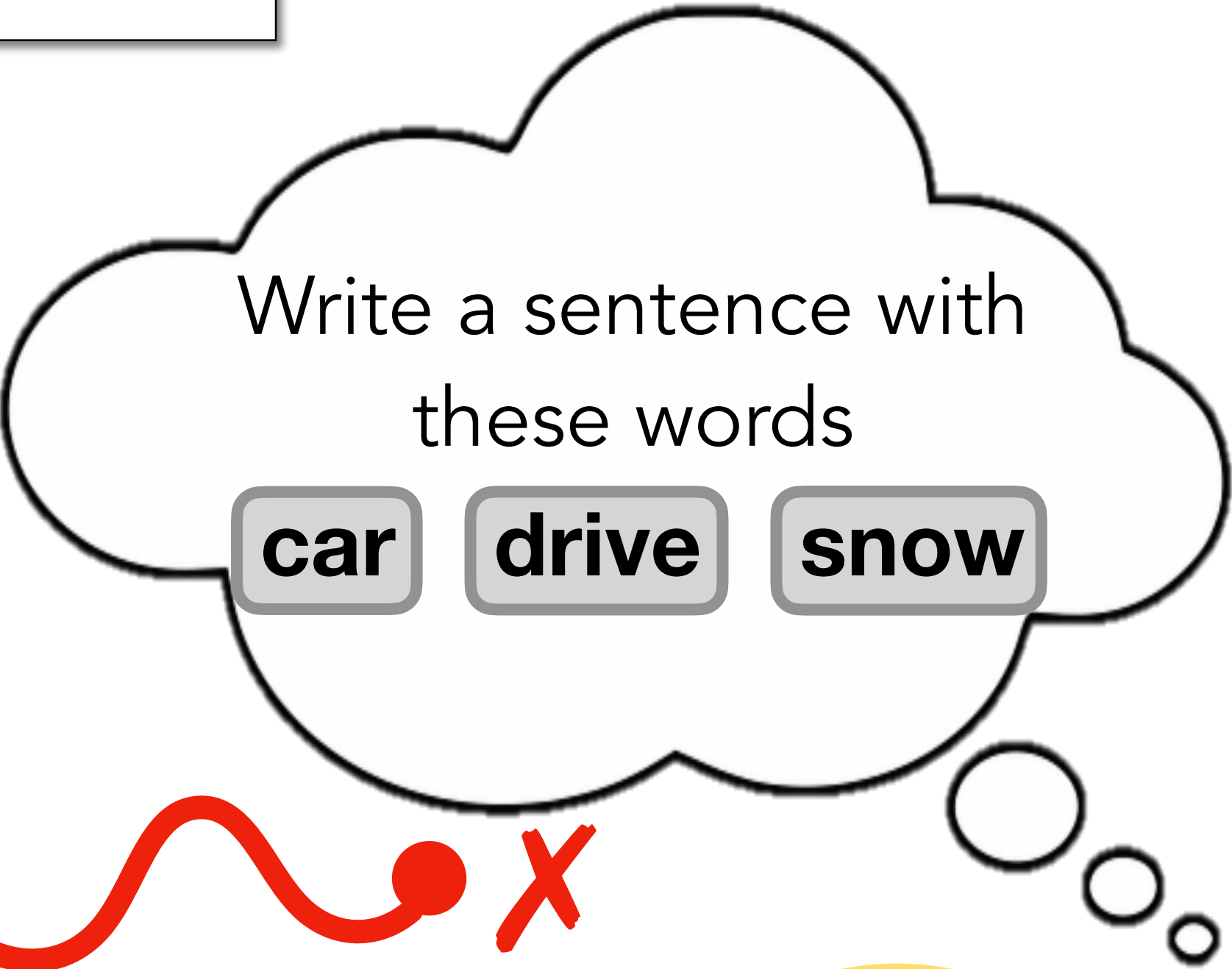
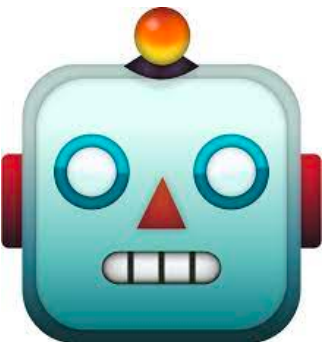


NeuroLogic Decoding

$$\text{score } s = \log P_{\theta}(y_t | y_{<t}) + \alpha' \sum_{i=1}^m C_i$$

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

Off-the-Shelf GPT2



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

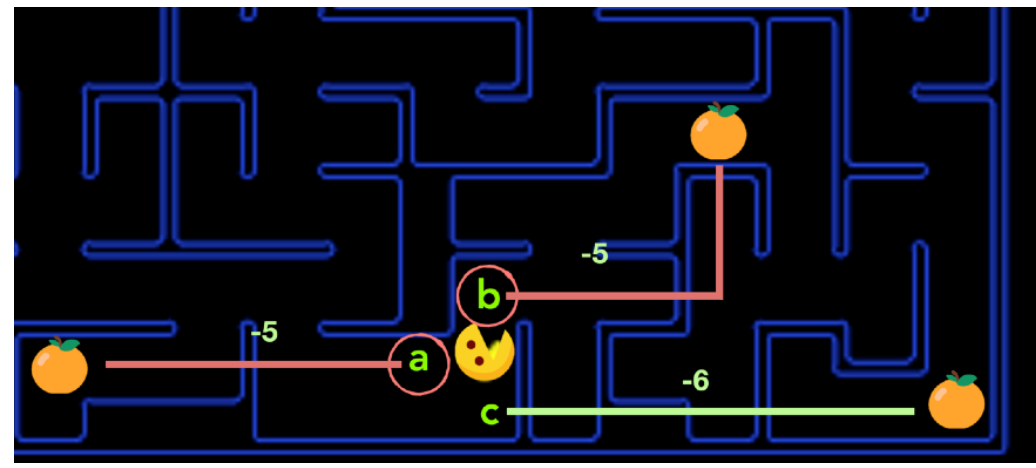
summer
 $p(w | \text{past}) = 0.4$

winter
 $p(w | \text{past}) = 0.2$
snow ✓

day



A* Search



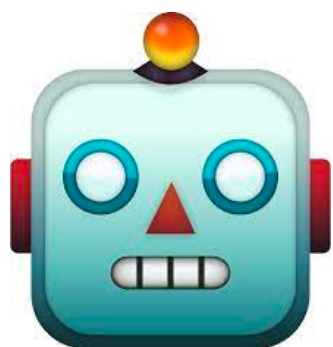
NeuroLogic A* ESQUE

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

← A* 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

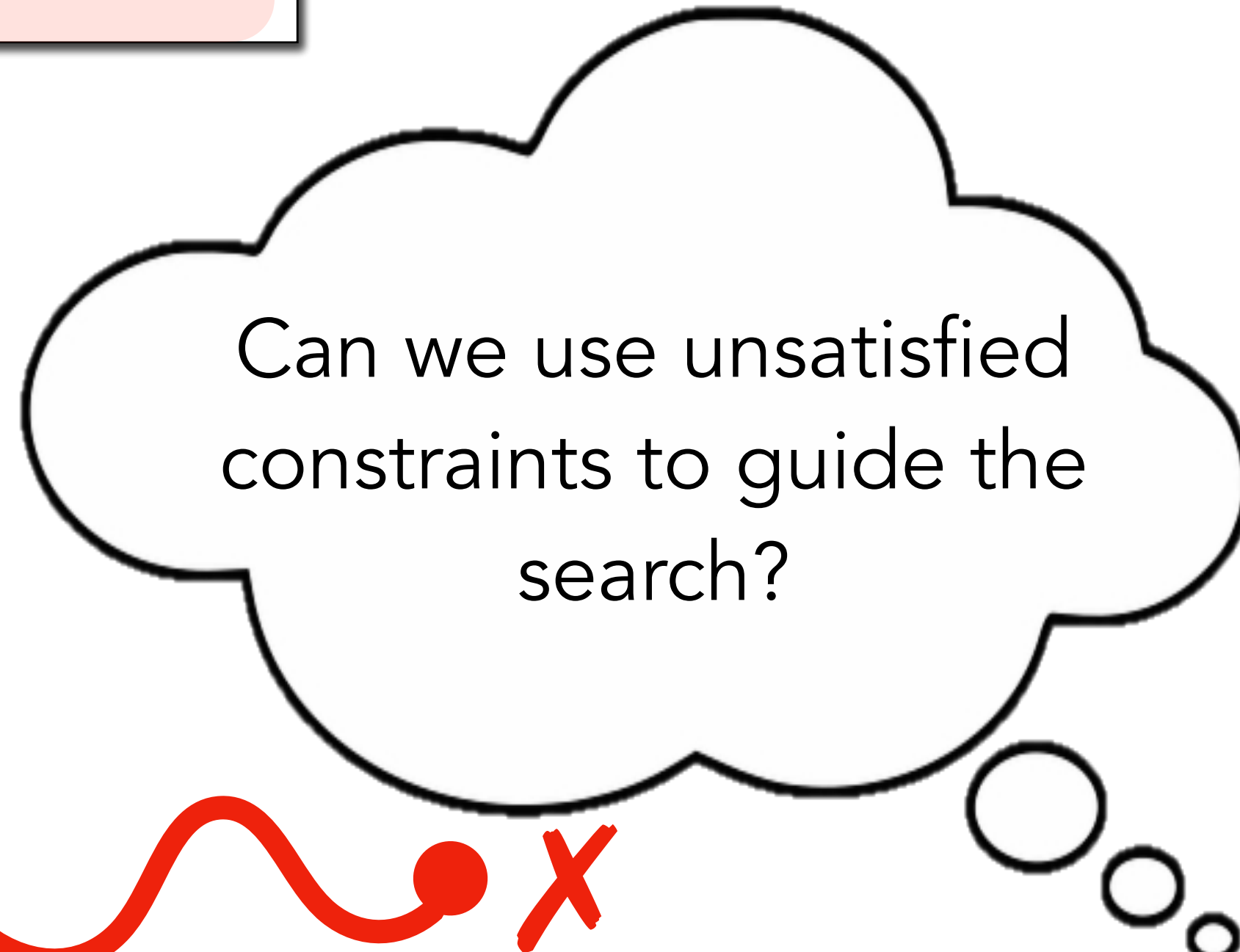
$p(w | \text{past}) = 0.4$

winter

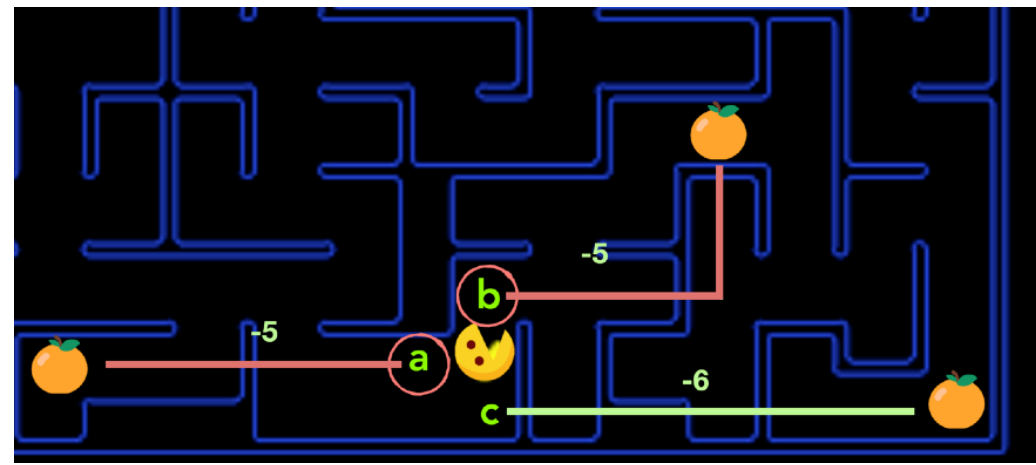
$p(w | \text{past}) = 0.2$

day

snow ✓



A* Search



NeuroLogic A[★]ESQUE

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

← A[★] Heuristic

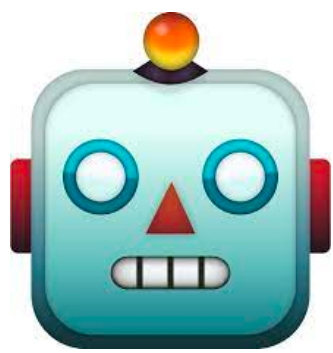
$D_1(\text{car}) \wedge D_2(\text{drive}) \wedge D_3(\text{snow})$

greedy look-ahead

$$y_{t' \in [1, k]} = \arg \max_{y \in \mathcal{Y}} P_{\theta}(y | \mathbf{y}_{<t'})$$

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

Off-the-Shelf GPT2



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

summer

winter

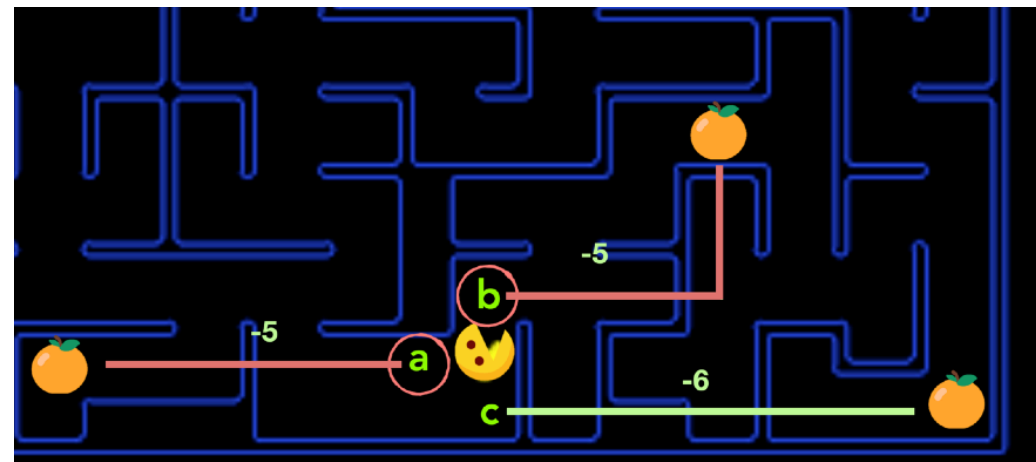
day

$$\max \left(\begin{matrix} \uparrow \\ \text{P}(\text{snowflake}) \end{matrix} \right)$$

$$\max \left(\begin{matrix} \uparrow \\ \text{P}(\text{snowflake}) \end{matrix} \right)$$

$$\max \left(\begin{matrix} \uparrow \\ \text{P}(\text{snowflake}) \end{matrix} \right)$$

A* Search



NeuroLogic A[★] ESQUE

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

← A[★] Heuristic

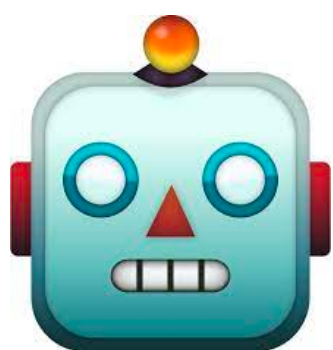
$D_1(\text{car}) \wedge D_2(\text{drive}) \wedge D_3(\text{snow})$

beam look-ahead

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

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

Off-the-Shelf GPT2



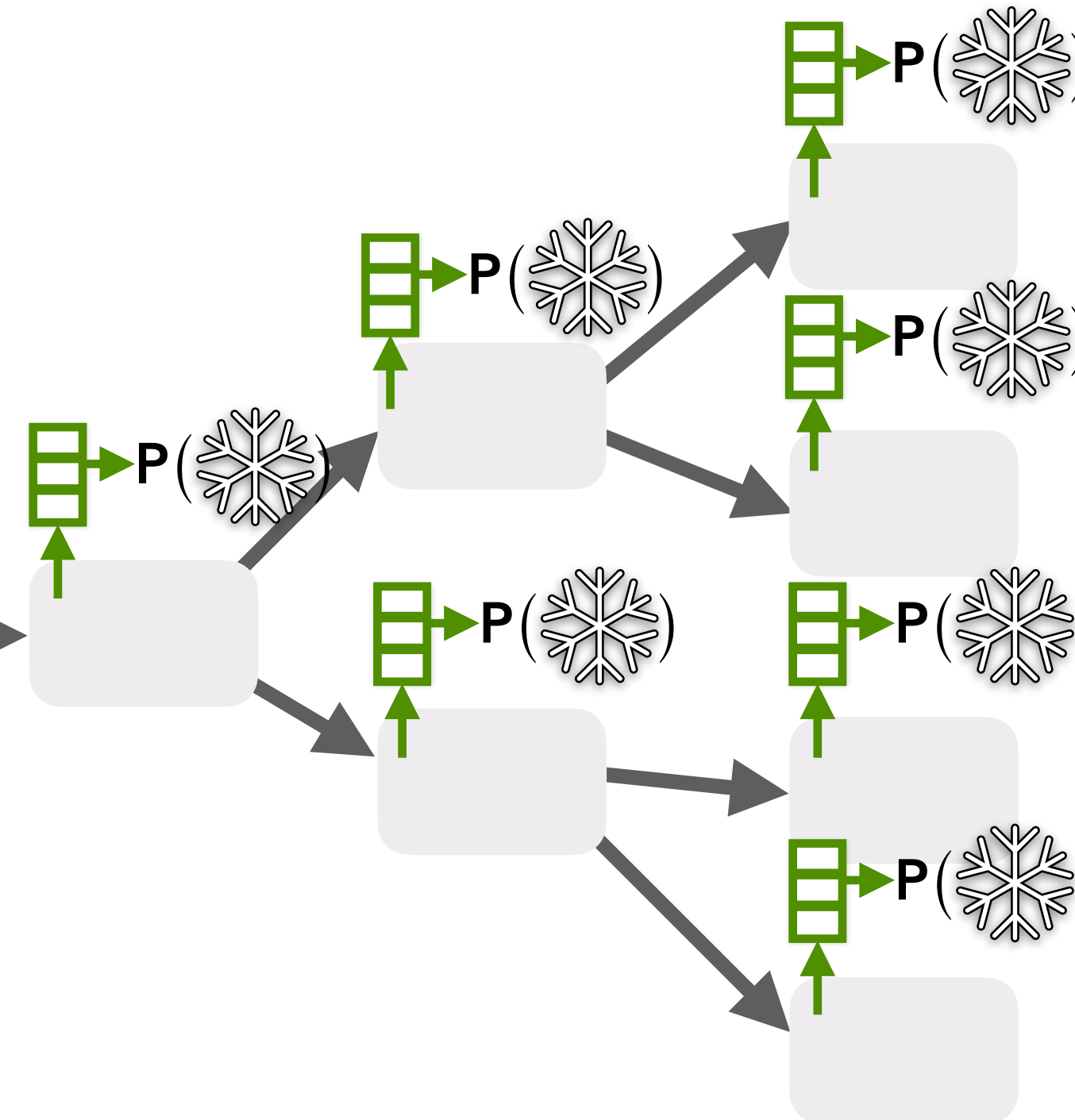
I **drive** my **car** during the

summer

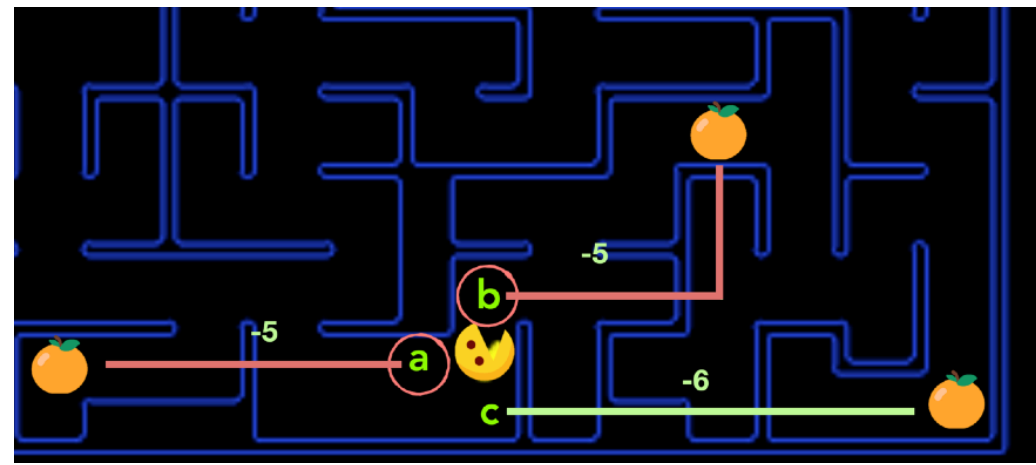
winter

day

max



A* Search



NeuroLogic A[★]ESQUE

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

← A[★] Heuristic

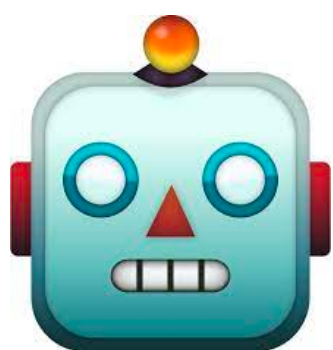
$D_1(\text{car}) \wedge D_2(\text{drive}) \wedge D_3(\text{snow})$

sampling look-ahead

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

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

Off-the-Shelf GPT2



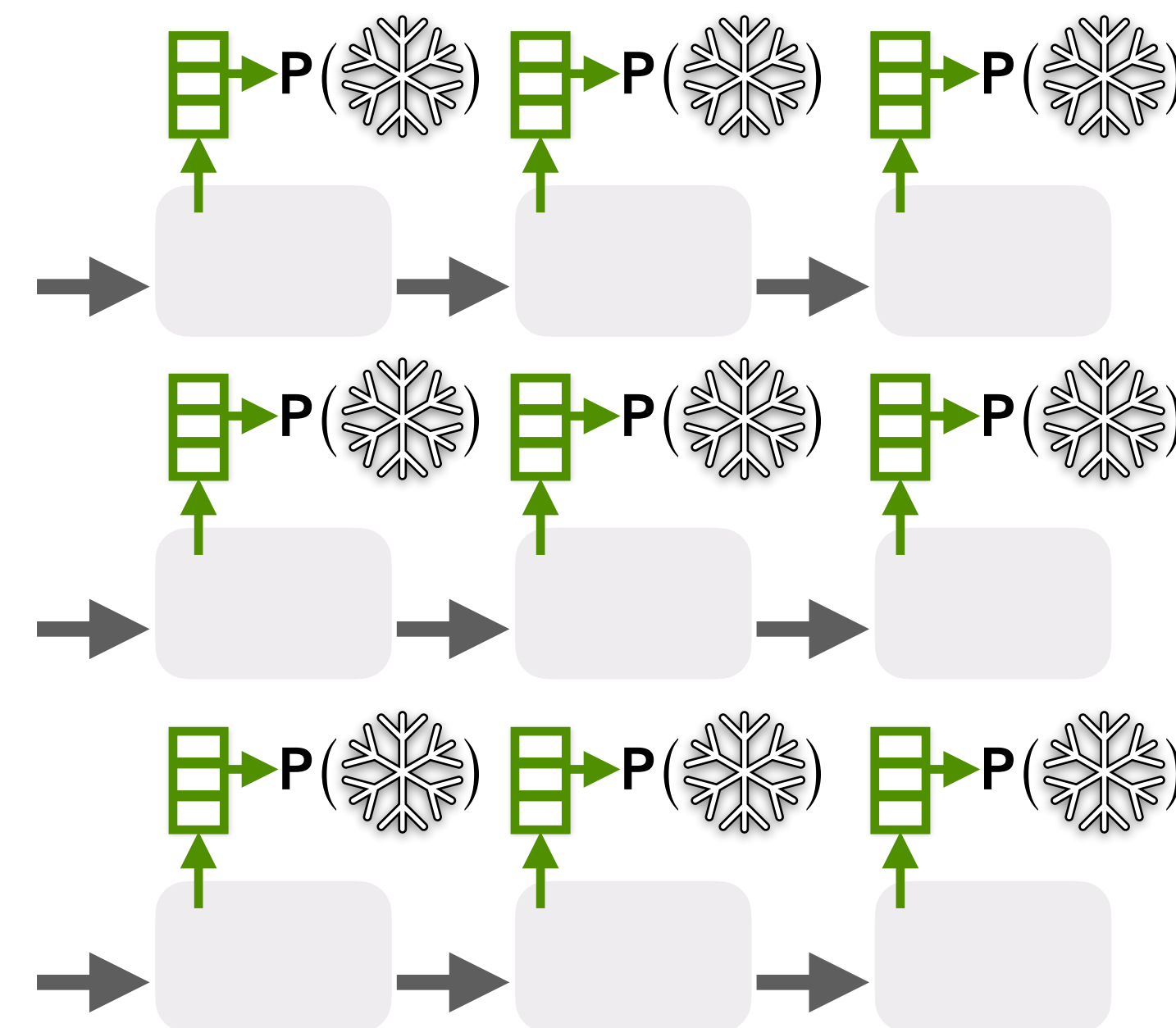
I **drive** my **car** during the

summer

winter

day

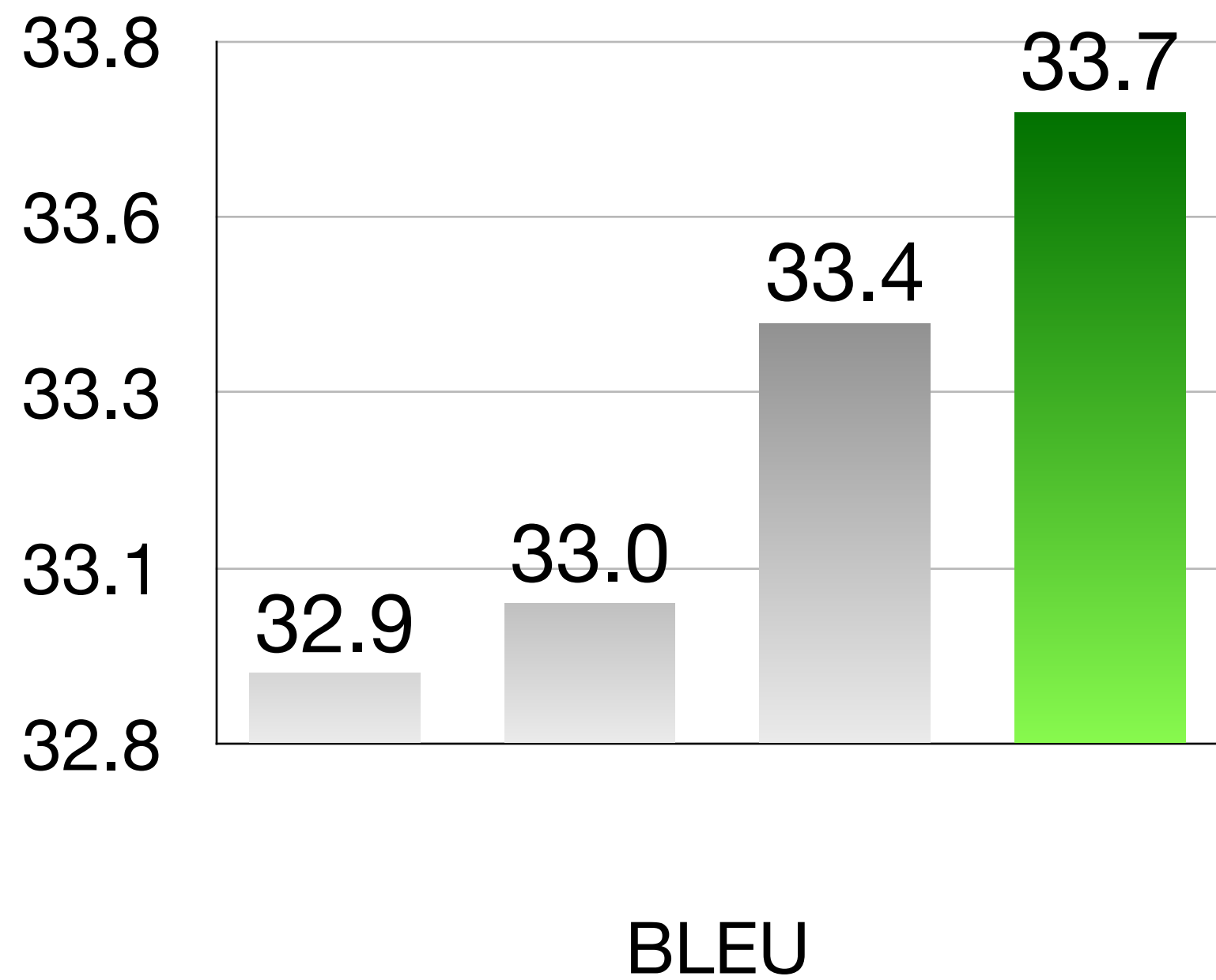
sample
max



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

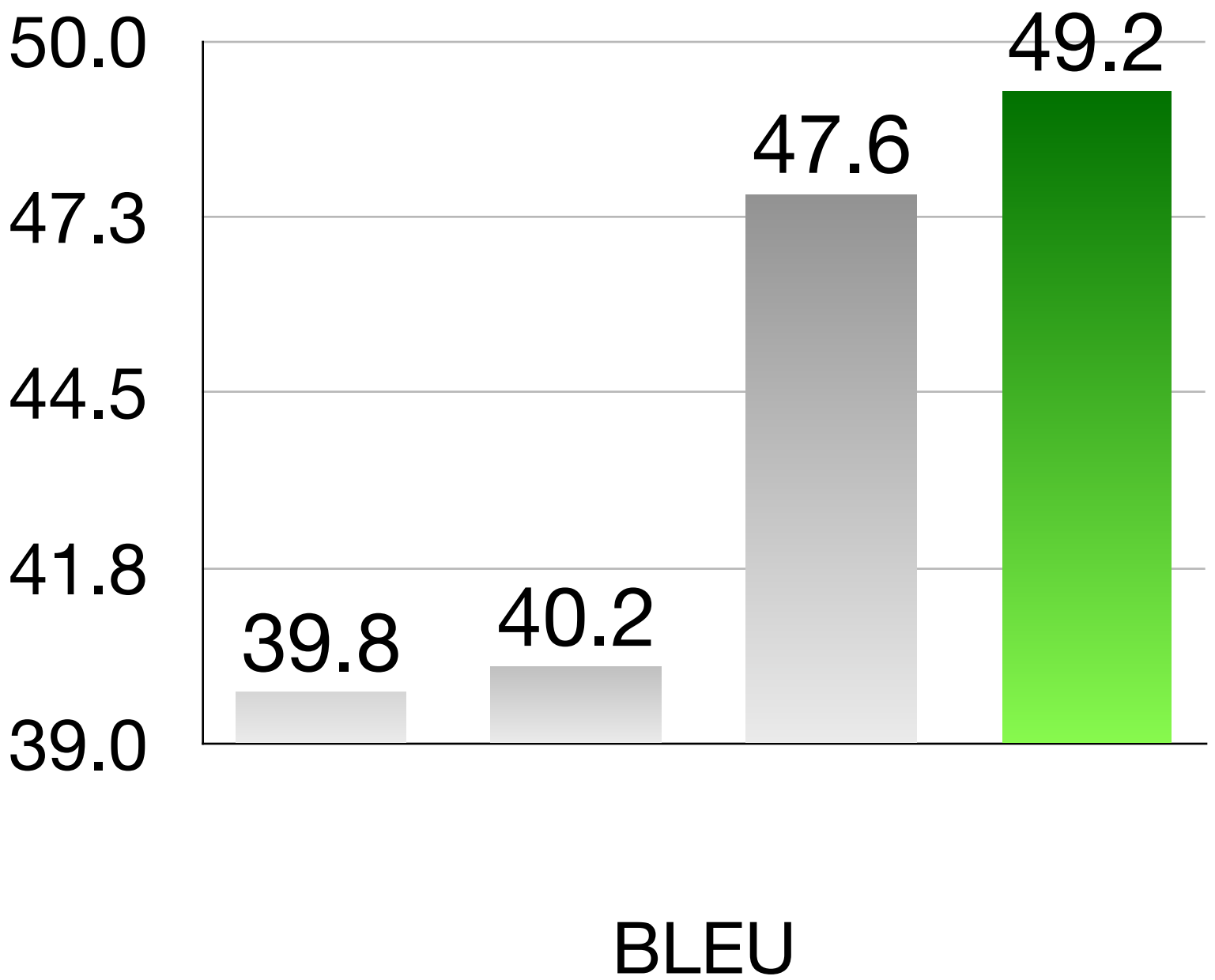
Constrained MT (Dinu et al., 2019)

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



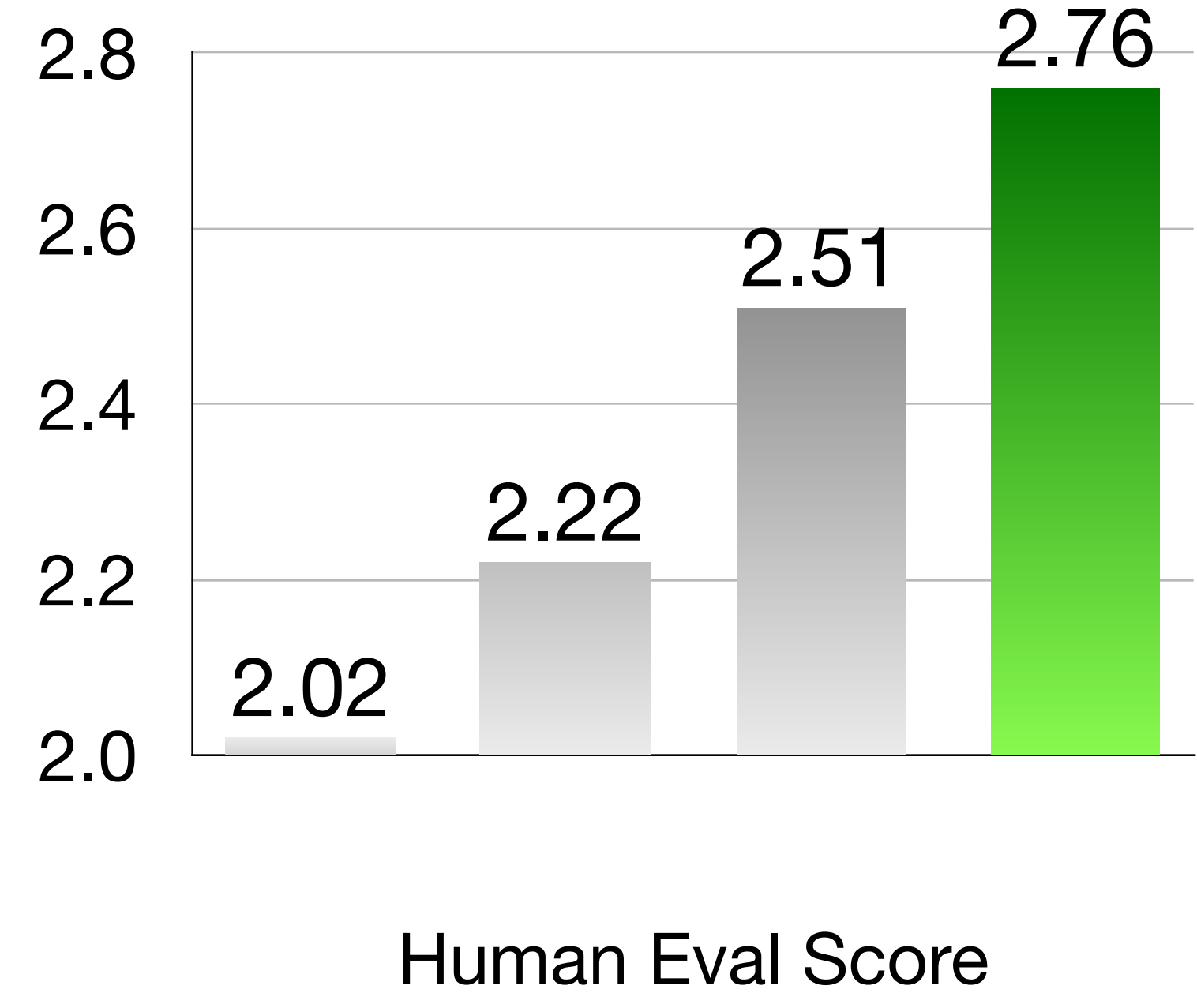
Few-Shot E2ENLG (Chen et al., 2020)

- 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



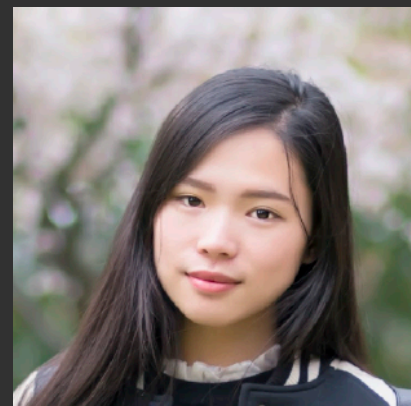
Keisuke
Sakaguchi



Ronan
Le Bras



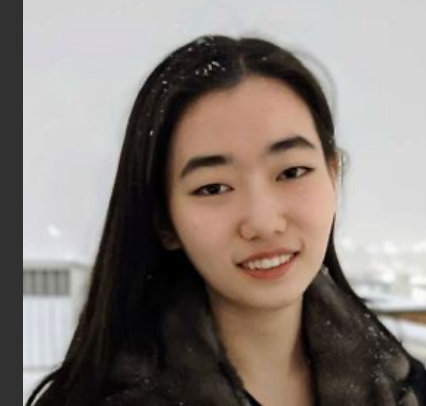
Lianhui
Qin



Peter
West



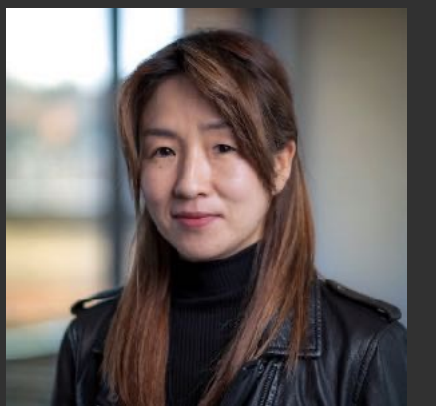
Ximing
Lu



Doug
Downey



Yejin
Choi



12D2: Neuro-Symbolic Generic Induction

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

12D2: Neuro-Sy

Style Constraints

$(\text{count}(\text{function_words}) = 1) \wedge (\text{count}(\text{co})$

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

Ximing Lu^{††} ♥ Sean Welleck^{††} ♥ Peter West[†]
Liwei Jiang^{††} Jungo Kasai^{††} Daniel Khashabi[†] Ronan Le Bras[†]

NEUROLOGIC DECODING: (Un)supervised Neural Text Generation with Predicate Logic Constraints

Ximing Lu^{††} Peter West^{††} Rowan Zellers^{††}
Ronan Le Bras[†] Chandra Bhagavatula[†] Yejin Choi^{††}



Bicycle

Concept

Prompt
Construction

A bicycle can

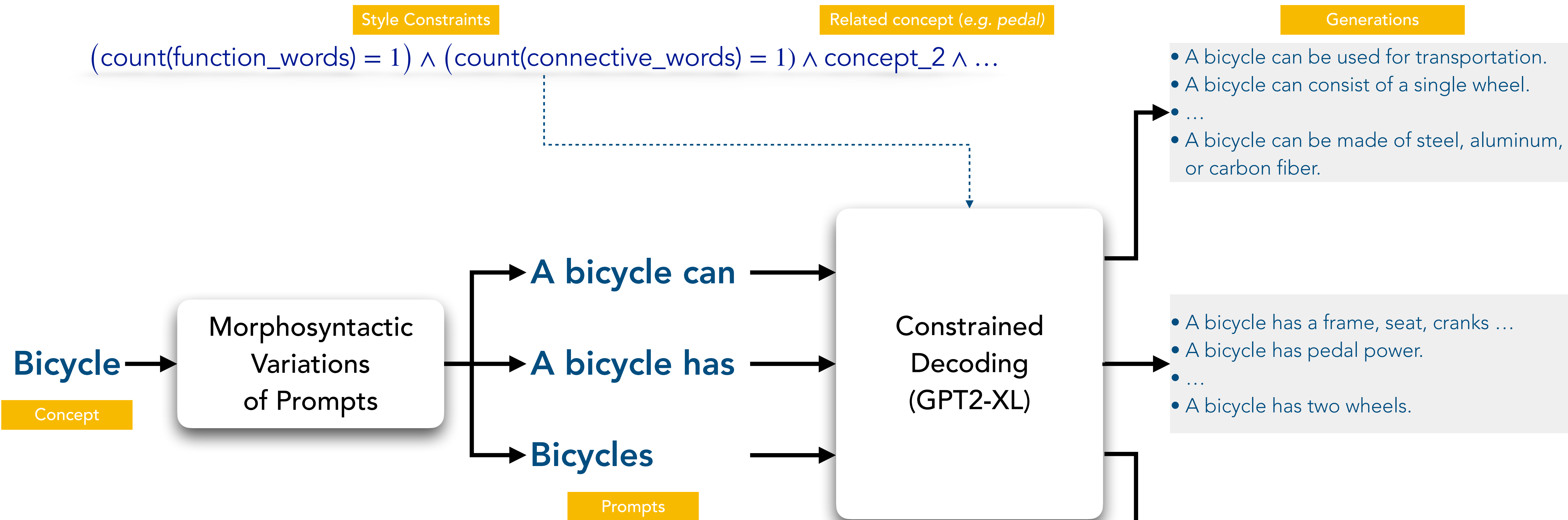
A bicycle has

Bicycles

Prompts

Constrained
Decoding
(GPT2-XL)

2D2: Neuro-Symbolic Generic Induction



NEUROLOGIC DECODING:
(Un)supervised Neural Text Generation with Predicate Logic Constraints

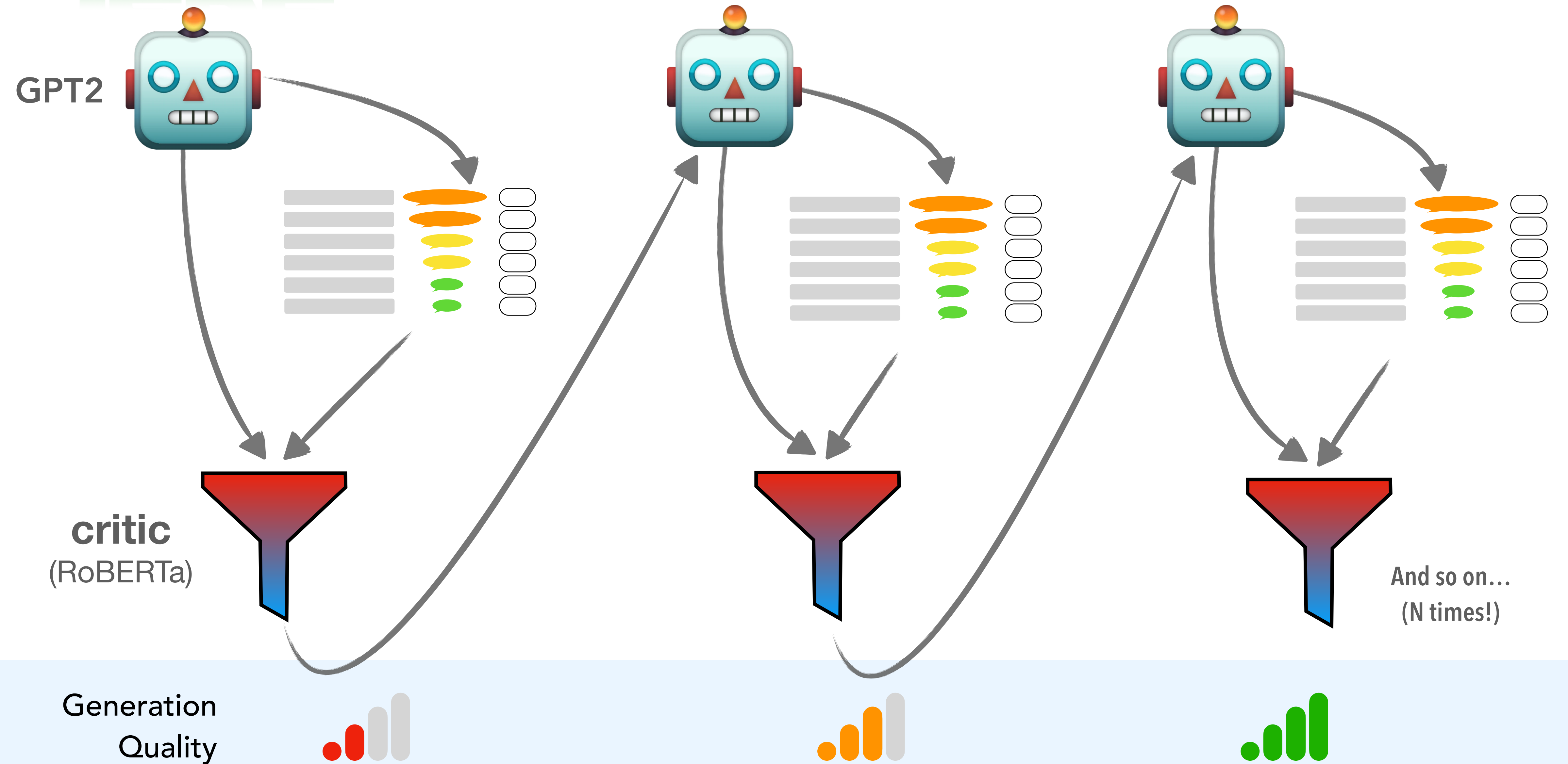
NEUROLOGIC A★esque Decoding:
Constrained Text Generation with Lookahead Heuristics

Ximing Lu^{††} ♥ Sean Welleck^{††} ♥ Peter West[†]
Liwei Jiang^{††} Jungo Kasai^{††} Daniel Khashabi[†] Ronan Le Bras[†]
Lianhui Qin[†] Youngjae Yu[†] Rowan Zellers[†] Noah A. Smith^{††} Yejin Choi^{††}

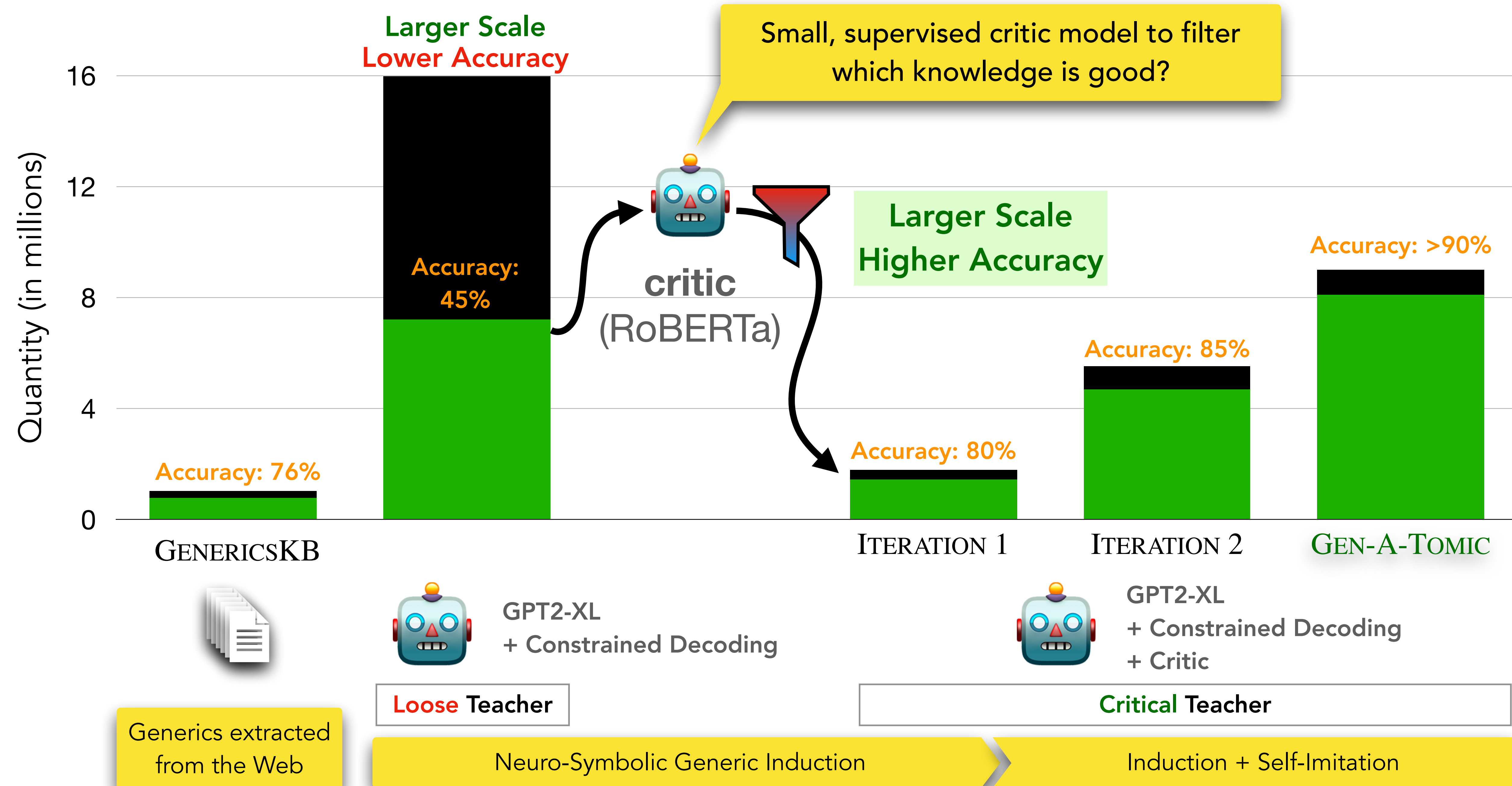


Low Quality!

12D2: Critic Filtering & Self-Imitation

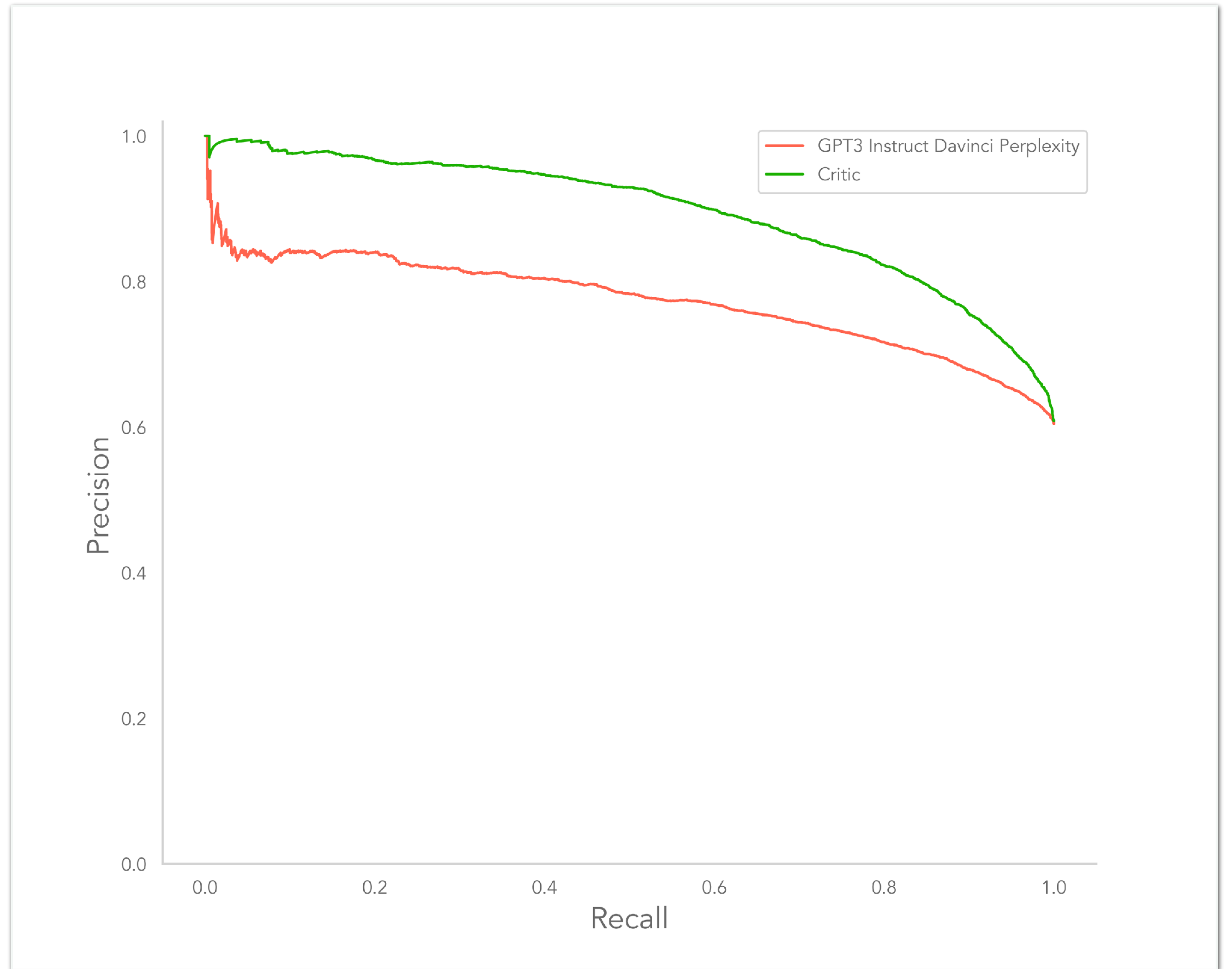


Does **12D2** Produce **high-quality** knowledge?



👋 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

Passed the bar exam



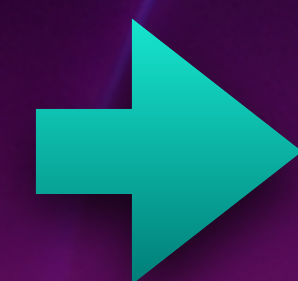
Existential risk



AI not yet as smart as a dog



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*¹

Affiliations:

¹Stanford University, Stanford, CA94305,

*Correspondence to: 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[♠][◇] **Ronan Le Bras**[♠] **Daniel Fried**[◇] **Yejin Choi**[♠][♡]

[♠]Allen Institute for AI, Seattle, WA, USA

[◇]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA

[♡]Paul G. Allen School of Computer Science, University of Washington, Seattle, WA, USA

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)

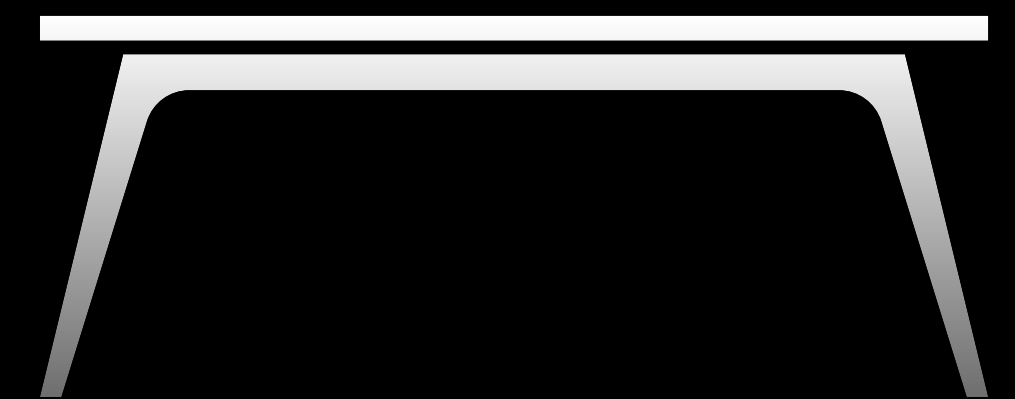
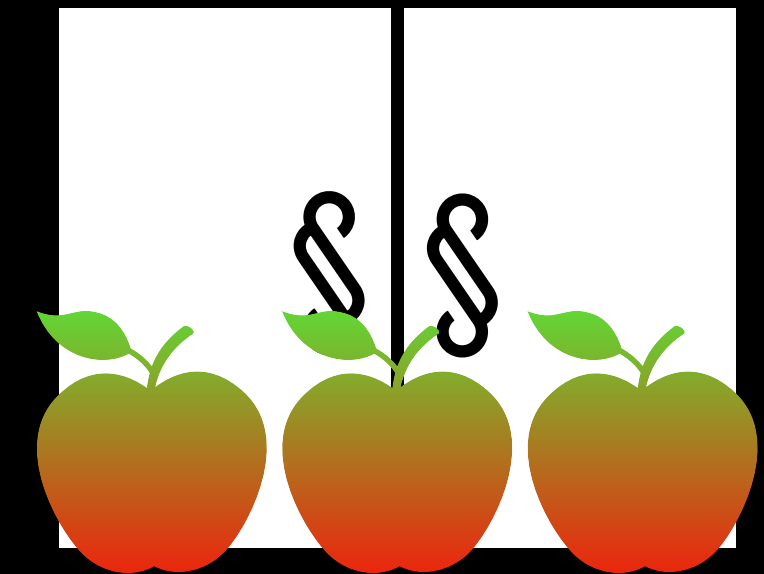
“theory of mind” test

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?



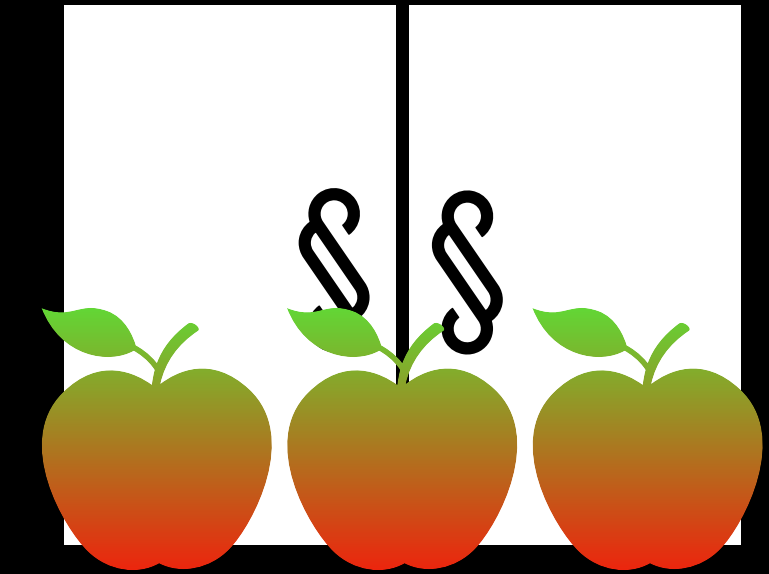
Circa 2022... (GPT-3)

“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



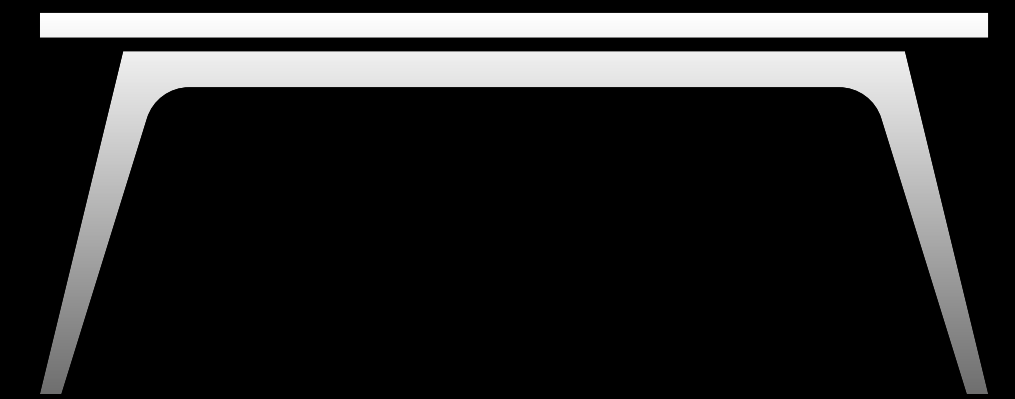
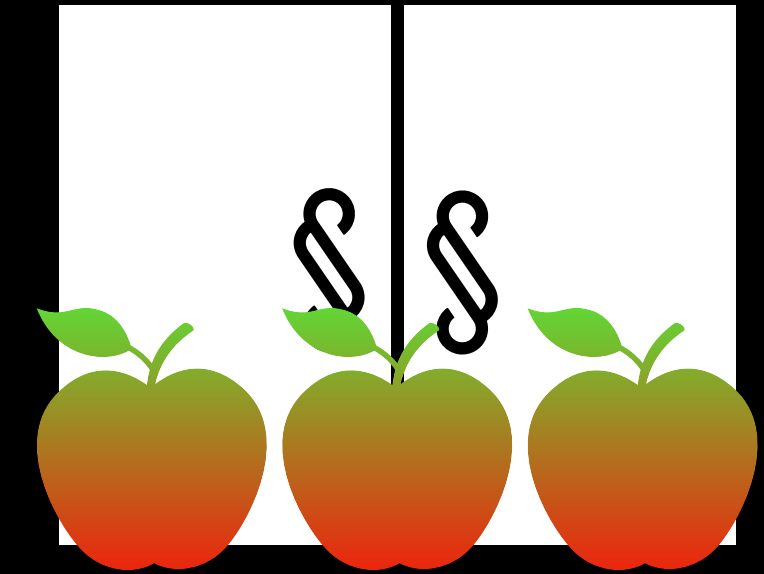
Circa 2023... (GPT-4)

“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



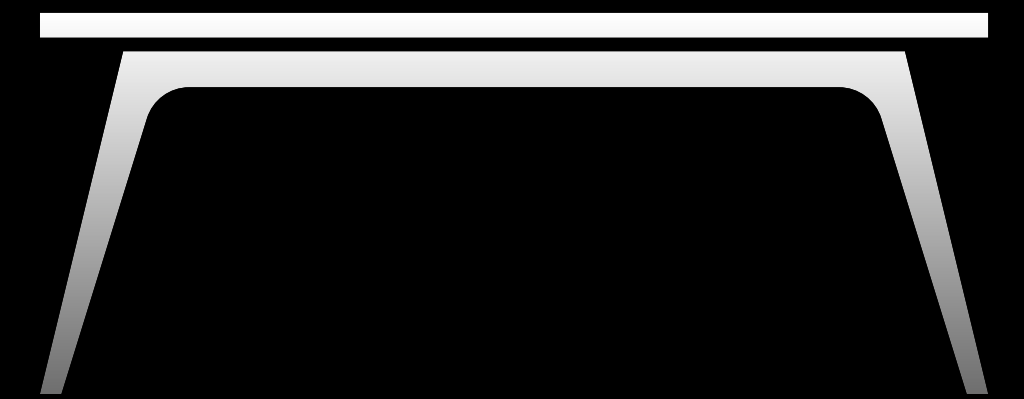
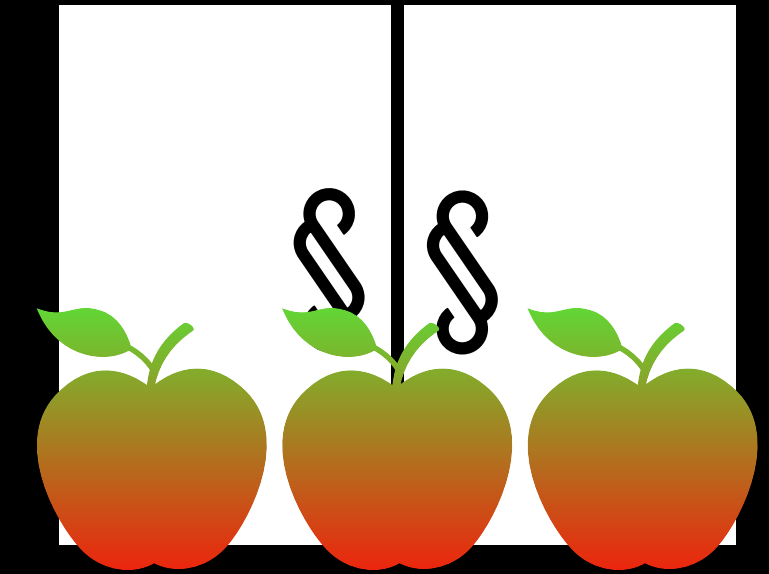
Circa 2023... (GPT-4)

“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 Sclar¹

Sachin Kumar²

Peter West¹

Alane Suhr³

Yejin Choi^{1,3}

Yulia Tsvetkov¹



*ACL 2023 *outstanding paper award**

GPT4 - 68%

Typical false-belief
ToM story:

1 room
2 people*
2 containers
1 object

GPT4 - 58%

Variant 1

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?



* with an extra distractor person (ToMi dataset)

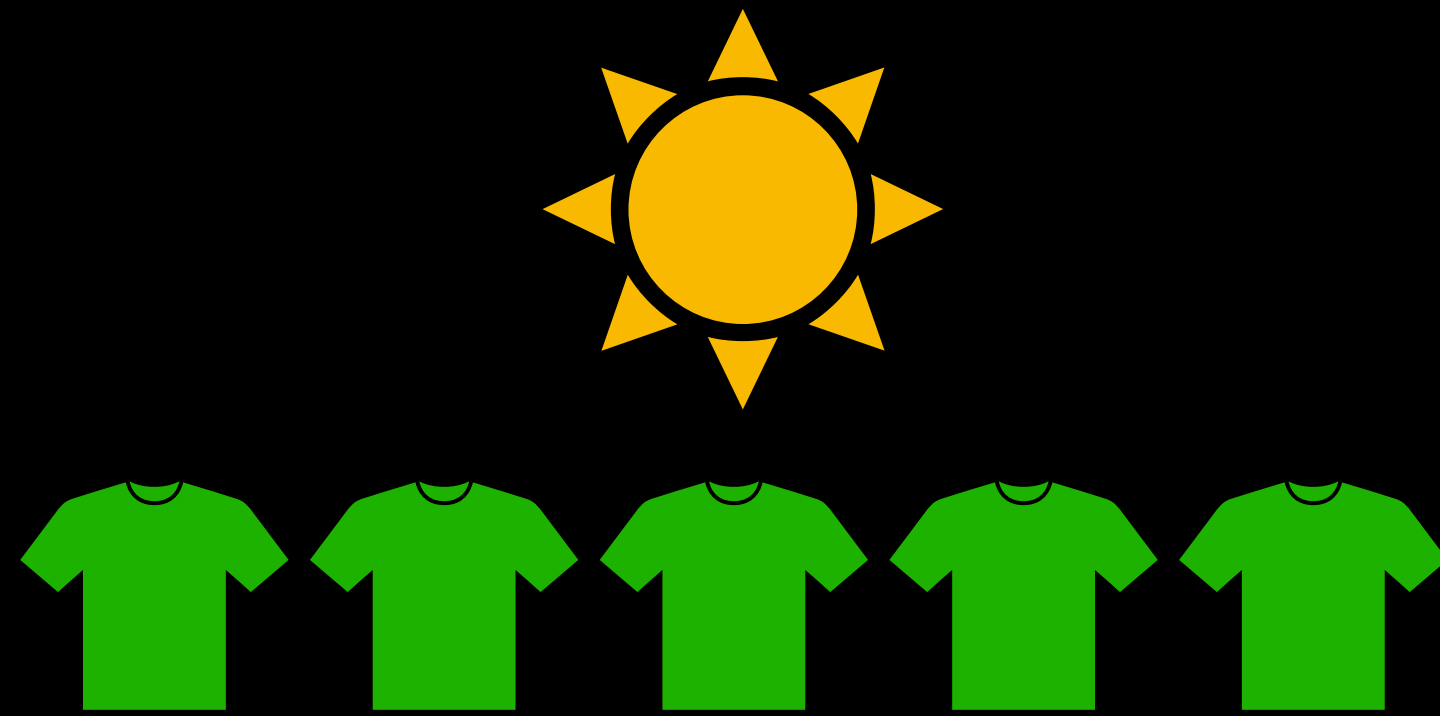


Why AI is incredibly smart and shockingly stupid

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 —

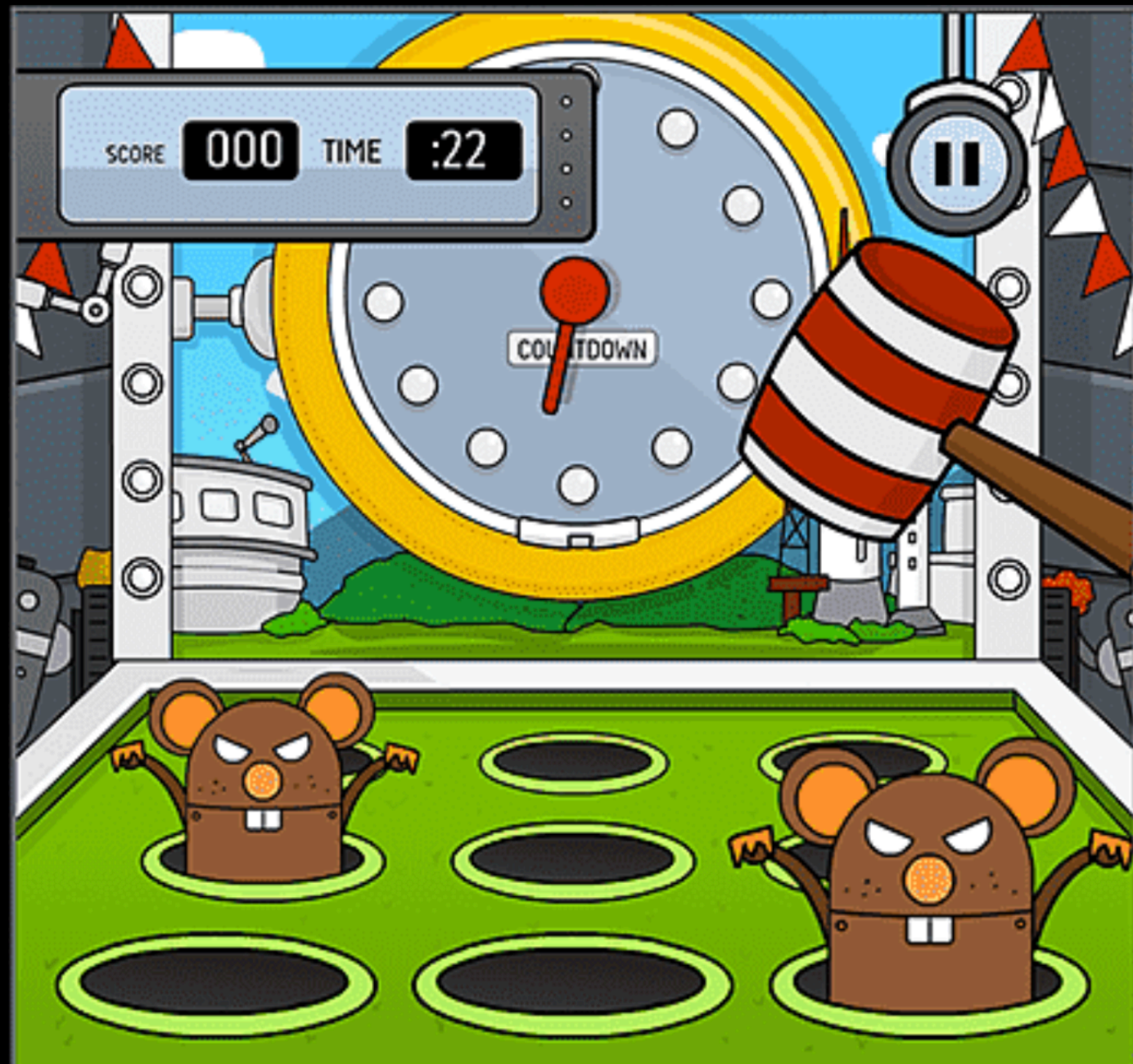


The drying time for clothes is typically dependent on factors like the type of cloth, the intensity of the sunlight, humidity, wind, and so on. However, if we are assuming that the drying time scales linearly with the number of clothes, and that all other factors remain constant, we can calculate it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

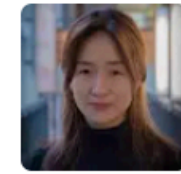
If it takes 10 hours to dry 5 clothes, then it takes 2 hours to dry 1 cloth (10 hours / 5 clothes = 2 hours/cloth).

So, it would take 40 hours to dry 20 clothes (20 clothes * 2 hours/cloth = 40 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 hours / 5 clothes = 2 hours/cloth).

So, it would take 40 hours to dry 20 clothes (20 clothes * 2 hours/cloth = 40 hours).

Commonsense Paradox

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

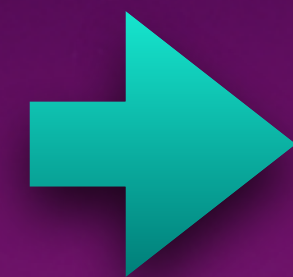
- Commonsense is trivial for humans, hard for machines
- Among humans, "common sense is not so common" — Voltaire
- LLMs do acquire a vast amount of commonsense knowledge
- Yet in some ways, "AI is worse than a dog" — Yann Lecun

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^{1,2,4} Eric Mintun² Nikhila Ravi^{1,2} Hanzi Mao² Chloe Rolland³ Laura Gustafson³
 Tete Xiao³ Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár⁴ Ross Girshick⁴
¹project lead ²joint first author ³equal contribution ⁴directional 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*² Gabriel Ilharco*¹ Alex Fang*¹ Jonathan Hayase¹ Georgios Smyrnis⁵
 Thao Nguyen¹ Ryan Marten^{7,9} Mitchell Wortsman¹ Dhruva Ghosh¹ Jieyu Zhang¹
 Eyal Orgad³ Rahim Entezari¹⁰ Giannis Daras⁵ Sarah Pratt¹ Vivek Ramanujan¹
 Yonatan Bitton¹¹ Kalyani Marathe¹ Stephen Mussmann¹ Richard Vencu⁶
 Mehdi Cherti^{6,8} Ranjay Krishna¹ Pang Wei Koh¹ Olga Saukh¹⁰ Alexander Ratner¹
 Shuran Song² Hannaneh Hajishirzi^{1,7} Ali Farhadi¹ Romain Beaumont⁶
 Sewoong Oh¹ Alexandros G. Dimakis⁵ Jenia Jitsev^{6,8}
 Yair Carmon³ Vaishaal Shankar⁴ Ludwig Schmidt^{1,6,7}



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 Schuhmann¹ §§^{oo} Romain Beaumont¹ §§^{oo} Richard Ven
Cade Gordon² §§^{oo} Ross Wightman¹ §§ Mehdi Cherti^{1,10}
Theo Coombes¹ Aarush Katta¹ Clayton Mullis¹ Mitchell Wo
Patrick Schramowski^{1,4,5} Srivatsa Kundurthy¹ Katherine Crowson
Ludwig Schmidt⁶ oo Robert Kaczmarczyk^{1,7} oo Jenia Jitsev^{1,10} oo

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

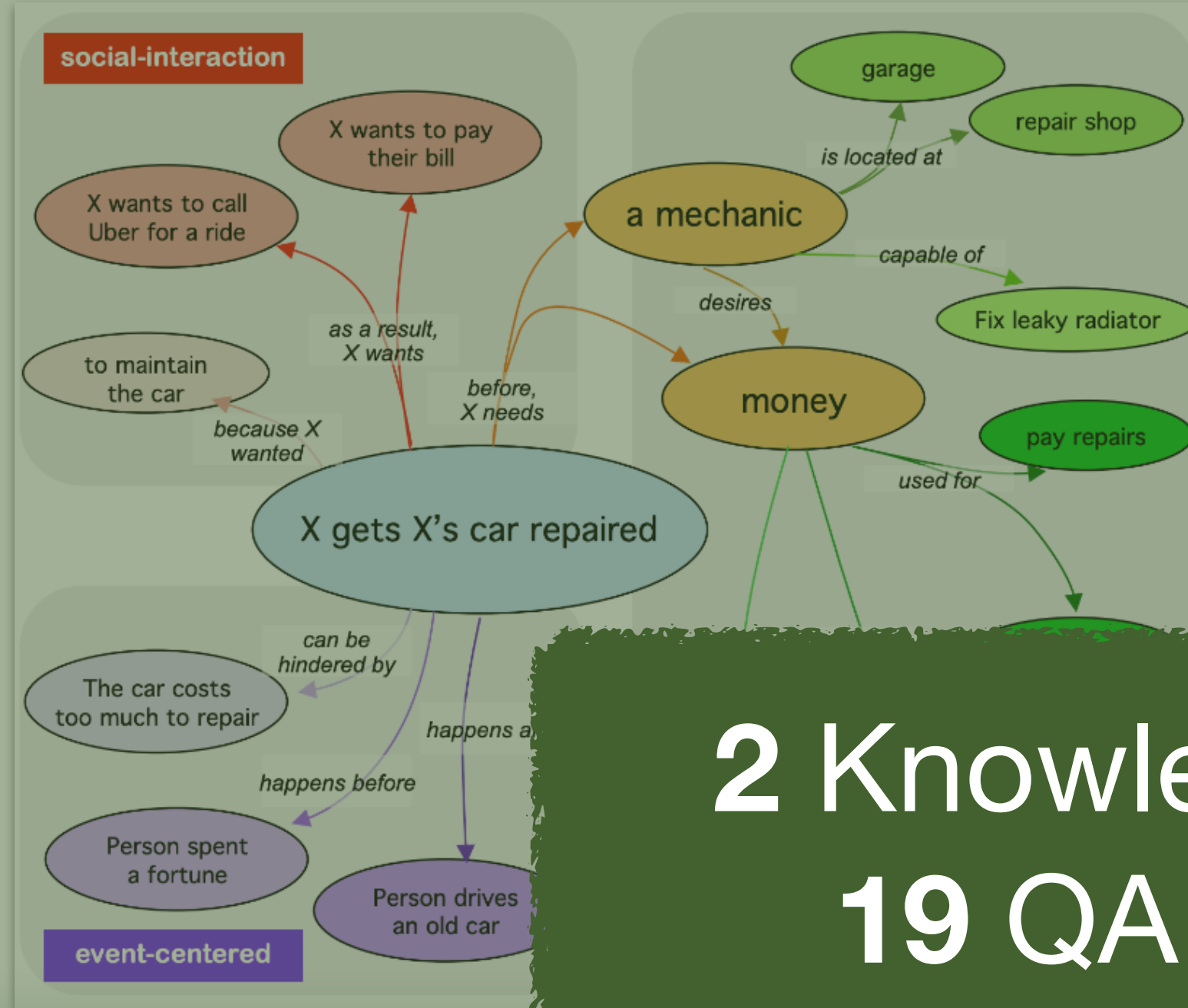
**Jiacheng Liu^{♡*} Wenya Wang^{♡*} Dianzhuo Wang[◇]
Noah A. Smith^{♡♠} Yejin Choi^{♡♠} Hannaneh Hajishirzi^{♡♠}**

Plausibility: 15%



A bird has four legs.

Atomic2020 [Hwang et al., 2021]



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

Original example

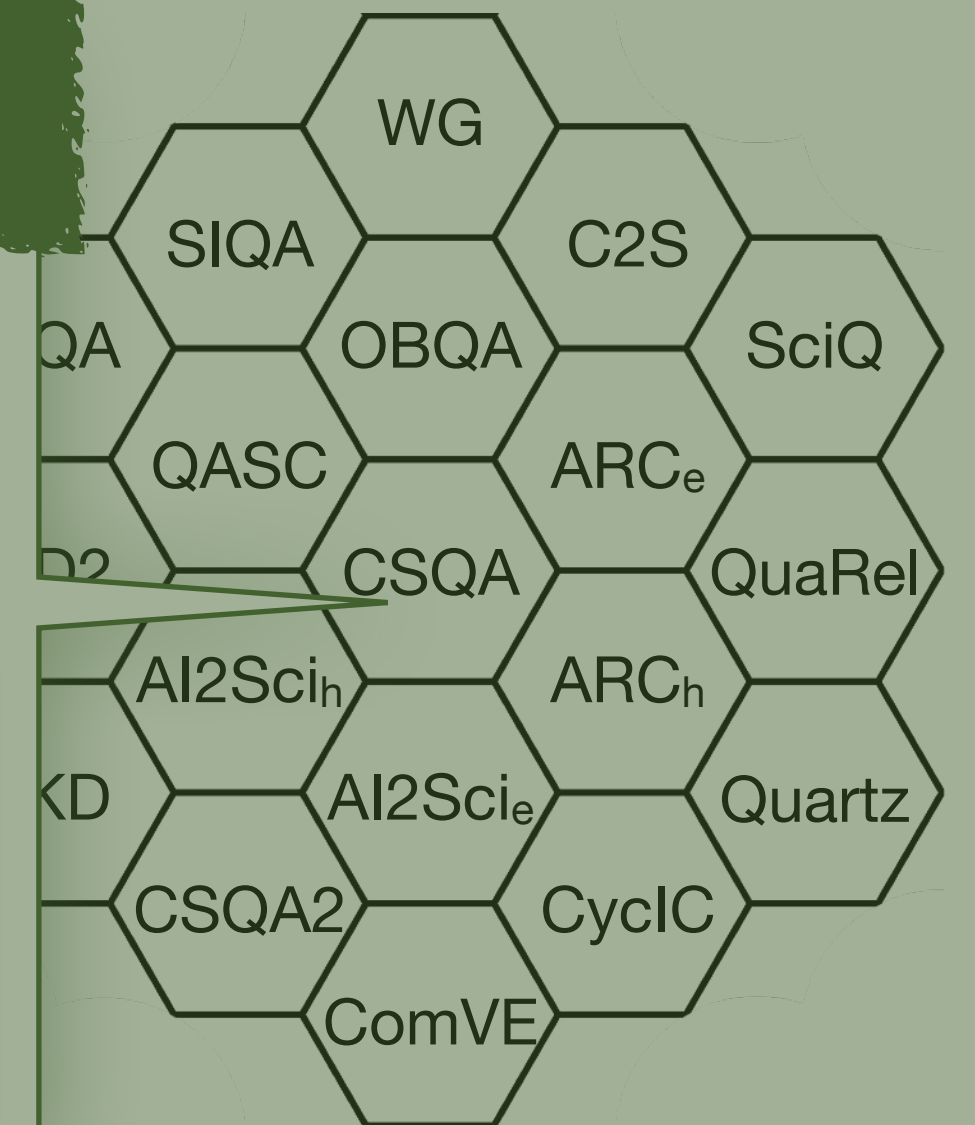
What would some
 (A) ungulate (B) bomber (C) body armor (D) tank (E) hat
 Answer: (C)

↓ Data Conversion

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)

Datasets



Solving Commonsense Benchmarks

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



Vera

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



3%

6%

93%

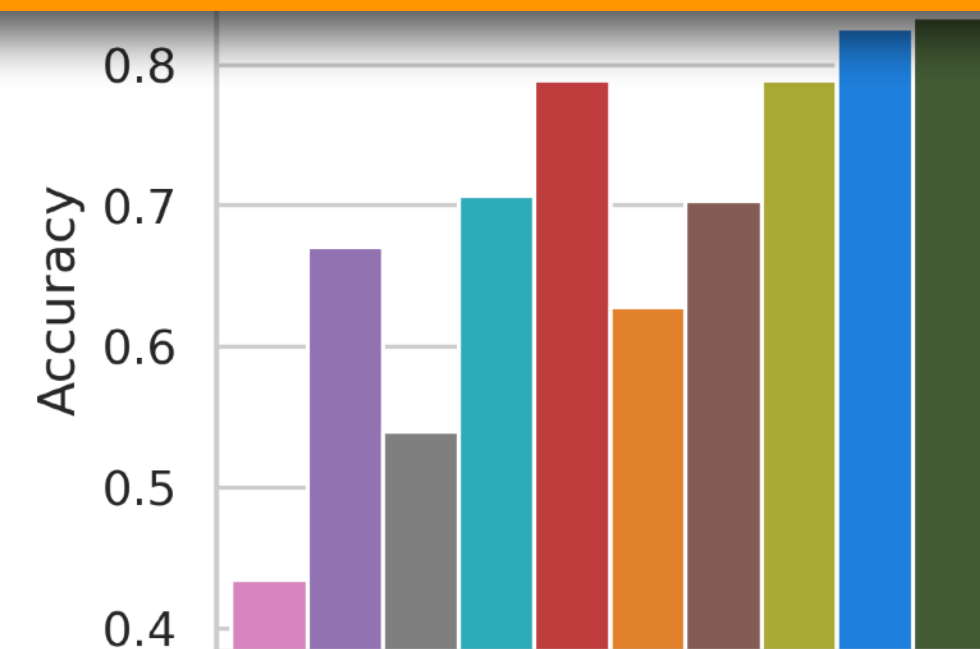
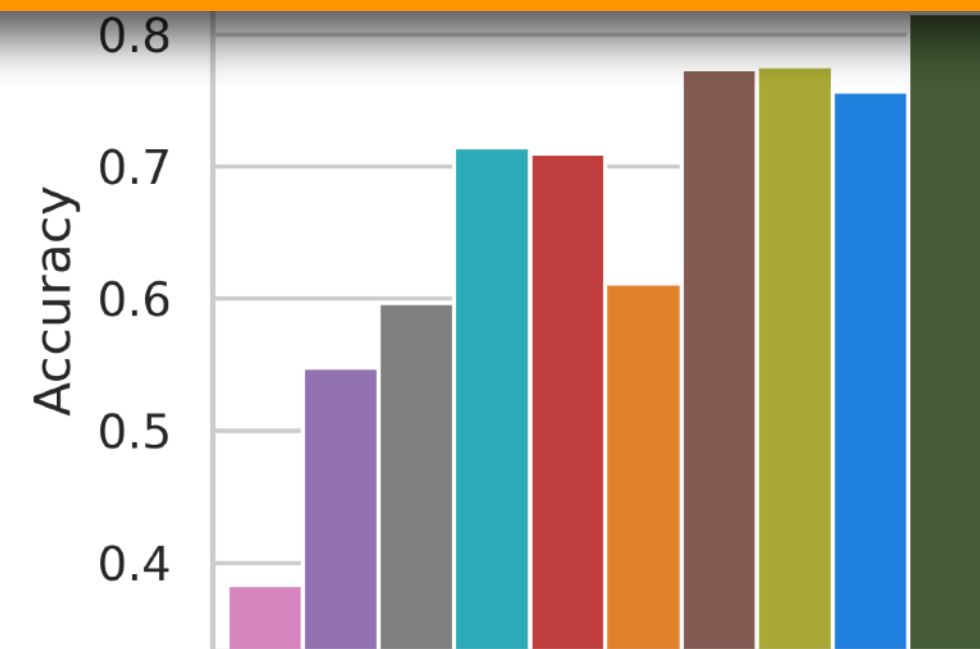
Best baseline is Flan-T5. ChatGPT and GPT-4 are worse.

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

EVALUATION (UNSEEN TYPE 1)	
WSC	multiple-choice (4)
COPA	
NumerSense	
PROST	
Spatial Commonsense	
EVALUATION (UNSEEN TYPE 2)	
SWAG	boolean
HellaSwag	
CODAH	
Story Cloze Test	
αNLI	
StrategyQA	
CREAK	

5 unseen (type 1) benchmarks
 Similar to seen benchmarks, but diagnostic datasets

8 unseen (type 2) benchmarks
 The tasks are a bit further from commonsense verification





Thanks!