2050: An AI Odyssey: Dark Matter of Intelligence

Yejin 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 will 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?

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

Few-shot prompting & Instruction tuning?
NeRF? Diffusion? Transformers?
Autonomous driving? Cleaning? Plumbing? Baby seating?
LLMs (or LVMs?) as prior?
Scaling laws no more?

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

“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 understand the known fundamentals”

Philipp von Jolly

“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?
Multiply 999 by 867.

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

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

```
  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
```

Therefore, 999 multiplied by 867 equals 866,733.
Multiplying 999 by 867.

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

Best paper I've read in a while. No excessive hype, clear question, rigorous empirical evaluation. Highly recommended.

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
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*)

Multiply 999 by 867.

GPT2-XL

The product is 865,233.

The correct answer is 66,233

*90M examples include 1x1, 2x1, 2x2, 3x1, 3x2, 3x3, 4x1, 4x2, 4x3, 4x4 multiplication problem sizes, 14 epochs, 80:10:10 train/dev/test

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*)

Multiply 999 by 866.

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 7.

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. ....

The final answer is 865,134.

Accuracy increases to 92% on 3x3 multiplication!!

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

The final answer is 16,695,134.

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. ....

The final answer is 3,305,251,134.
Wait, didn’t previous work said “transformers absolutely can learn true multi-step algorithms in-context”???

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

Transformers absolutely can learn and generalize to arbitrary problem.

<table>
<thead>
<tr>
<th>Method</th>
<th>Subtraction</th>
<th>Multiplication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithmic prompt</td>
<td>65.6%</td>
<td>79.7%</td>
</tr>
<tr>
<td>Best available baseline</td>
<td>16.7%</td>
<td>5.5%</td>
</tr>
</tbody>
</table>

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

We investigate the **fundamental limits** of achieving **full mastery** of the task rather than incremental improvements.
How about fine-tuning GPT3 on scratchpad?

<table>
<thead>
<tr>
<th>Problem size</th>
<th># Examples</th>
<th>Cost* USD</th>
</tr>
</thead>
<tbody>
<tr>
<td>2x2</td>
<td>8100</td>
<td>$744</td>
</tr>
<tr>
<td>3x2</td>
<td>81K</td>
<td>$7440</td>
</tr>
<tr>
<td>3x3</td>
<td>810K</td>
<td>$74,404</td>
</tr>
<tr>
<td>4x3</td>
<td>8.1M</td>
<td>$744,040</td>
</tr>
<tr>
<td>4x4</td>
<td>81M</td>
<td>$7.4 M</td>
</tr>
<tr>
<td>5x4</td>
<td>810M</td>
<td>$70.4 M</td>
</tr>
<tr>
<td>5x5</td>
<td>8.1B</td>
<td>$700 M</td>
</tr>
</tbody>
</table>

*Cost for 4 epochs with text-davinci-003
How about fine-tuning GPT3 on scratchpad?

<table>
<thead>
<tr>
<th>Cost* USD</th>
<th># Examples</th>
<th>Problem size</th>
</tr>
</thead>
<tbody>
<tr>
<td>$744</td>
<td>8100</td>
<td>2x2</td>
</tr>
<tr>
<td>$7440</td>
<td>81K</td>
<td>3x2</td>
</tr>
<tr>
<td>$74,404</td>
<td>810K</td>
<td>3x3</td>
</tr>
<tr>
<td>$744,040</td>
<td>8.1M</td>
<td>4x3</td>
</tr>
<tr>
<td>$7.4 M</td>
<td>81M</td>
<td>4x4</td>
</tr>
<tr>
<td>$70.4 M</td>
<td>810M</td>
<td>5x4</td>
</tr>
<tr>
<td>$700 M</td>
<td>8.1B</td>
<td>5x5</td>
</tr>
</tbody>
</table>

*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.

*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, 4x4, etc.

**Why does this happen? Can we understand Transformers’ behaviour via computation graphs?**
function multiply (a[1:p], b[1:q]):
for i = q to 1
    carry = 0
    for j = p to 1
        summands[i] = digits
        product = summands[q+1-i] \cdot 10^{i-1}
        return product
for j = p to 1
    t = a[j] \ast b[i]
    t += carry  (only if j != p)
    digits[j] = t \mod 10
    carry = t \div 10
    summands[i] = digits
product = \sum_{i=1}^{q} summands[q+1-i] \cdot 10^{i-1}
return product
Model Performance Decreases as Graph Complexity Increases

$9 \times 7 = 63$

$49 \times 7 = 343$

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

<table>
<thead>
<tr>
<th>graph width</th>
<th>graph depth</th>
<th>avg. parallelism</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>1.3</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>2.8</td>
</tr>
</tbody>
</table>

No. digits
- $9 \times 7 = 63$
- $49 \times 7 = 343$

GPT4 zero-shot
What is the correlation between a model generating a correct output and having seen relevant subgraphs during training?
Detect subgraphs already seen during training: if giant 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 some of its ancestors have incorrect values.  
**Restoration Error:** v has a correct value but is derived from an incorrect computation.

---

Five-shot GPT4 – Multiplication

- Fully Correct
- Local Error
- Propagation Error
- Restoration Error

Fine-tuned GPT3 – Multiplication

- Fully Correct
- Local Error
- Propagation Error
- Restoration Error
Transformers’ performance will rapidly decay with increased task complexity.
By and large, the prior work was based on weaker LLMs, thus some might have wondered with extreme-scale, these problems magically go away.
Let’s step back...

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

That’s exactly the point! Relationally, 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?

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

GPT-2
Low-quality, small models

GPT-3

High-quality, small models

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

<table>
<thead>
<tr>
<th>Name</th>
<th>University</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arnav Gudibande*</td>
<td>UC Berkeley</td>
<td><a href="mailto:arnavg@berkeley.edu">arnavg@berkeley.edu</a></td>
</tr>
<tr>
<td>Xinyang Geng</td>
<td>UC Berkeley</td>
<td><a href="mailto:young.geng@berkeley.edu">young.geng@berkeley.edu</a></td>
</tr>
<tr>
<td>Eric Wallace*</td>
<td>UC Berkeley</td>
<td><a href="mailto:ericwallace@berkeley.edu">ericwallace@berkeley.edu</a></td>
</tr>
<tr>
<td>Hao Liu</td>
<td>UC Berkeley</td>
<td><a href="mailto:hao.liu@berkeley.edu">hao.liu@berkeley.edu</a></td>
</tr>
<tr>
<td>Charlie Snell*</td>
<td>UC Berkeley</td>
<td><a href="mailto:csnell22@berkeley.edu">csnell22@berkeley.edu</a></td>
</tr>
<tr>
<td>Pieter Abbeel</td>
<td>UC Berkeley</td>
<td><a href="mailto:pabbeel@berkeley.edu">pabbeel@berkeley.edu</a></td>
</tr>
<tr>
<td>Sergey Levine</td>
<td>UC Berkeley</td>
<td><a href="mailto:svlevine@berkeley.edu">svlevine@berkeley.edu</a></td>
</tr>
<tr>
<td>Dawn Song</td>
<td>UC Berkeley</td>
<td><a href="mailto:dawnsong@berkeley.edu">dawnsong@berkeley.edu</a></td>
</tr>
</tbody>
</table>
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‡ †
‡ Bras‡ † Ximing Lu‡ † Sean Welleck‡ † Yejin Choi †‡

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

Aligning Sentences

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*, Chun-Liang Li†, Chih-Kuan Yeh‡, Hootan Nakhost‡,
Yasuhsa Fujii†, Alexander Ratner†, Ranjay Krishna†, Chen-Yu Lee†, Tomas Pfister‡
†University of Washington, ‡Google Cloud AI Research, *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” ✨
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

GPT-3

high-quality, small models

VS
Symbolic Knowledge Distillation
From General Language Models to Commonsense Models
— NAACL 2022 —
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?
GPT-2

Pool of candidate pairs

Filters

Entailment filter
remove non-factual summaries using NLI

Length filter
remove too long summaries

Diversity filter

Filters for Summarization

Summarization Dataset

T5-large
Filters for Paraphrasing

- GPT-2
- T5-large

Pool of candidate pairs

Paraphrasing Dataset

Filters

- Bidirectional Entailment filter
- Abstractness filter
- Length filter
- Diversity filter
Generalizes to both tasks by simply re-defining the filters!

Train a single model capable of both tasks!
Self-Distillation yields better dataset, stronger task model

- GPT-2
  - Pool of candidate pairs
  - Filters
    - Summ / Para Dataset
      - T5-large
- Flan-T5
  - Better Dataset
  - Self-Distillation
    - Better Task Model
      - T5-ImpDistill
        - 770M LM capable of both controllable summarization + paraphrasing, distilled purely from < 2B LMs
      - DimSum+
        - 3.4M samples for sentence summarization + paraphrasing, spanning news / reddit / bio domains
"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.

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

![Bar graph showing comparisons between different models on various metrics.](image-url)
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!
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

Long-form QA

Open-Ended Generation

Machine Translation

Dialogue
What is the mass of Jupiter?

Generate a question containing all of the given words.

Words: Jupiter, Mercury, Venus, mass

Language Model (GPT3)

Neural Language Models

GPT-3 (175B)
Megatron-Turing NLG (530B)
Megatron-LM (8.3B)
Turing-NLG (17.2B)
GPT-2 (1.5B)
BERT-Large (340M)
ELMo (94M)

COMMONGEN (Liu et al 2020)

missing keywords
Neural Language Models

What is the mass of Jupiter?
Best-first Search
Dijkstra's

What is the mass of Jupiter?

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

Neural Language Models

Search Algorithms in Classical AI

A* Search
MinMax Search
Monte Carlo Tree Search

Best-first Search
Dijkstra's
What is the mass of Jupiter?
NeuroLogic Decoding in a Nut Shell

Constraints: $\left(\text{cowboy} \land \text{dog} \land \left(\text{play music} \lor \text{plays music}\right) \land \left(\text{catch} \lor \text{catches}\right)\right)$

<table>
<thead>
<tr>
<th>search tree</th>
<th>likelihood</th>
<th>clauses</th>
<th>score</th>
<th>select</th>
<th>notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>runs</td>
<td>0.18</td>
<td></td>
<td>$0.18 + 0.1 \times 0 = 0.18$</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>dog</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>catches</td>
<td>0.12</td>
<td></td>
<td>$0.12 + 0.1 \times 0 = 0.12$</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>plays</td>
<td>0.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>eats</td>
<td>0.20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>talks</td>
<td>0.16</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>talks</td>
<td>0.15</td>
<td></td>
<td>$0.15 + 0.1 \times 0 = 0.15$</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>cowboy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>plays</td>
<td>0.11</td>
<td></td>
<td>$0.11 + 0.1 \times \frac{1}{2} = 0.16$</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>catches</td>
<td>0.09</td>
<td></td>
<td>$0.09 + 0.1 \times 0 = 0.09$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pruning step:
- $\bigcirc$ denotes failure in top-$\alpha$ filtering in term of likelihood,
- $\bigcirc$ denotes failure in top-$\beta$ filtering in term of number of satisfied clauses

Grouping step:
- $1 \ 2 \ 3 \ 4$ denotes candidate groups based on the shared set of irreversibly satisfied clauses

Selecting step:
- $\bigcirc$ denotes the top-1 candidate within each group ranked by score function. Among these candidates, we select $\checkmark$ the top-$k$ ones to fill in the next beam.

$$s = P_0(y_t | y_{<t}) + \lambda \cdot \max_{D(a_t,y) \in \text{state S1}} \left| \hat{a}_t \right|$$
NeuroLogic Decoding in a Nut Shell

— it’s a logic-guided search algorithm

<table>
<thead>
<tr>
<th>Reversible</th>
<th>Irreversible</th>
</tr>
</thead>
<tbody>
<tr>
<td>reversible satisfaction</td>
<td>reversible unsatisfaction</td>
</tr>
<tr>
<td>irreversible satisfaction</td>
<td>irreversible unsatisfaction</td>
</tr>
</tbody>
</table>

four states of clause satisfaction:

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

<table>
<thead>
<tr>
<th>States</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>( \neg D_1 \lor \neg D_2 \lor D_3 )</td>
</tr>
<tr>
<td>0 0 0</td>
<td>( \neg D_4 \lor \neg D_5 )</td>
</tr>
<tr>
<td>1 0 0</td>
<td>( \neg D_1 \lor \neg D_2 \lor D_3 )</td>
</tr>
<tr>
<td>1 0 0</td>
<td>( \neg D_1 \lor \neg D_2 \lor D_3 )</td>
</tr>
</tbody>
</table>

- violate all negative literals
- satisfy one positive literal
- violate all negative literals
- satisfy one positive literal
Unsupervised NeuroLogic outperforms supervised approaches on smaller networks outperforms supervised approaches on larger networks!
(Un)supervised Neural Text Generation with Predicate Logic Constraints
—NAACL 2021—

— 🏆 Best Method Paper Award at NAACL 2022 🏆 —

Constrained Text Generation with Lookahead Heuristic
NeuroLogic Decoding

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

\[ D_1(\text{car}) \land D_2(\text{drive}) \land D_3(\text{snow}) \]

Write a sentence with these words:

- car
- drive
- snow

Off-the-Shelf GPT2

I \textcolor{green}{\textbf{drive}} my \textcolor{green}{\textbf{car}} during the

- \text{summer}
- \text{winter}
- \text{day}
I drive my car during the summer.

\( p(w|past) = 0.4 \)

I drive my car during the winter.

\( p(w|past) = 0.2 \)
NeuroLogic \( \star \) ESQUE

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

Can we use unsatisfied constraints to guide the search?

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

Off-the-Shelf GPT2

I drive my car during the

- \( \text{summer} \) \( p(w | \text{past}) = 0.4 \)
- \( \text{winter} \) \( p(w | \text{past}) = 0.2 \)

\( \star \) Heuristic
I drive my car during the 

Off-the-Shelf GPT2

NeuroLogic A* ESQUE

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

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

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

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

beam look-ahead $Y_{t \in [1:k]} = \arg \text{topk}_{y \in \mathcal{Y}} P_\theta(y | y_{<t})$

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

Off-the-Shelf GPT2

I drive my car during the

summer

winter

day

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

\[
y_{r\in[1,k]} \sim P_\theta(y | y_{<r})
\]

\[
P_\theta(D_i(a) | Y_{\leq t+k}) = \max_{y \in Y} \max_{i \in [1,k]} P_\theta(y_{t+i:t+i+|a|} = a | y_{<t+i})
\]

**A* Heuristic**
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

Human Eval Score

BLEU

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>BLEU</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>MarianMT</td>
<td>32.8</td>
<td>33.0</td>
<td>33.3</td>
</tr>
<tr>
<td>Post and Vilar</td>
<td>33.0</td>
<td>33.4</td>
<td>33.7</td>
</tr>
<tr>
<td>NeuroLogic</td>
<td>33.3</td>
<td>33.6</td>
<td>33.8</td>
</tr>
<tr>
<td>NeuroLogic A* esque</td>
<td>33.7</td>
<td>34.1</td>
<td>34.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>BLEU</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>KGPT-Graph</td>
<td>39.8</td>
<td>40.2</td>
<td>40.6</td>
</tr>
<tr>
<td>KGPT-Seq</td>
<td>40.2</td>
<td>40.6</td>
<td>41.0</td>
</tr>
<tr>
<td>NeuroLogic</td>
<td>41.0</td>
<td>41.4</td>
<td>41.8</td>
</tr>
<tr>
<td>NeuroLogic A* esque</td>
<td>41.8</td>
<td>42.2</td>
<td>42.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>BLEU</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGMH</td>
<td>2.02</td>
<td>2.22</td>
<td>2.41</td>
</tr>
<tr>
<td>TSMH</td>
<td>2.22</td>
<td>2.41</td>
<td>2.60</td>
</tr>
<tr>
<td>NeuroLogic</td>
<td>2.41</td>
<td>2.60</td>
<td>2.76</td>
</tr>
<tr>
<td>NeuroLogic A* esque</td>
<td>2.60</td>
<td>2.76</td>
<td>2.92</td>
</tr>
</tbody>
</table>
I2D2: Inductive Knowledge Distillation with Neurologic and Self Imitation

— ACL 2023 —
“Generic statements” or “Generics” such as “birds can fly”
**NEUROLOGIC A*esque Decoding:**
Constrained Text Generation with Lookahead Heuristics

Ximing Lu††  Sean Welleck††  Peter West†
Liwei Jiang††  Juno 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†‡

**Neuro-Symbolic Generic Induction**

A bicycle can

A bicycle has

Bicycles

Bicycle

Prompt Construction

Concept

(count(function_words) = 1) \land (count(connective_words) = 1) \land \text{concept}_2 \land \ldots

Related concept (e.g. pedal)
Morphosyntactic Variations of Prompts

A bicycle can

A bicycle has

Bicycles

Constrained Decoding (GPT2-XL)

Style Constraints
(count(function_words) = 1) \land (count(connective_words) = 1) \land \text{concept}_2 \land ...

Related concept (e.g., pedal)

A bicycle can be used for transportation.

A bicycle can consist of a single wheel.

A bicycle can be made of steel, aluminum, or carbon fiber.

A bicycle has pedal power.

A bicycle has two wheels.

Bicycles should have a pedal.

Bicycles are also pedal.

Bicycles are a great way to commute.

Low Quality!

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

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

Ximing Li, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khoshaib, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, Yejin Choi

12D2: Neuro-Symbolic Generic Induction
I2D2: Critic Filtering & Self-Imitation

GPT2

critic (RoBERTa)

And so on... (N times!)

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

- **GenericsKB**
  - Accuracy: 76%
  - Generics extracted from the Web
  - Neuro-Symbolic Generic Induction

- **Iteration 1**
  - Accuracy: 80%
  - GPT2-XL + Constrained Decoding
  - Loose Teacher

- **Iteration 2**
  - Accuracy: 85%
  - GPT2-XL + Constrained Decoding + Critic
  - Critical Teacher

- **Gen-A-TOMIC**
  - Accuracy: >90%
  - Induction + Self-Imitation

**Summary:**
- Larger Scale vs. Higher Accuracy
- Small, supervised critic model to filter which knowledge is good?
Wait!!! Doesn’t GPT3 already have this knowledge? 🙃

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

![Graph showing precision vs. recall for GPT3 Instruct Davinci Perplexity and Critic.](image)
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*

Affiliations:
1Stanford University, Stanford, CA94305, USA
*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
Alice and Bob saw apples on the table in the kitchen.

Alice left the kitchen.

Bob moved the apples to the cabinet.
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?
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
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
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$^1$  Sachin Kumar$^2$  Peter West$^1$  Alane Suhr$^3$
Yejin Choi$^{1,3}$  Yulia Tsvetkov$^1$

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
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?

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?

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

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
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 Kirillov1,2,4 Eric Mintun2 Nikhila Ravi1,2 Hanzi Mao2 Chloe Rolland3 Laura Gustafson3 Tete Xiao3 Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár4 Ross Girshick4 1 project lead 2 joint first author 3 equal contribution 4 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*2 Gabriel Ilharco*1 Alex Fang*1 Jonathan Hayase1 Georgios Smyrnis5 Thao Nguyen1 Ryan Marten7,9 Mitchell Wortsman1 Dhruva Ghosh1 Jieyu Zhang1 Eyal Orgad3 Rahim Entezari10 Giannis Daras5 Sarah Pratt1 Vivek Ramamurthi2 Yonatan Bitton11 Kalyani Marathe1 Stephen Mussmann1 Richard Vencu6 Mehdi Chert6,8 Ranjay Krishna1 Pang Wei Koh1 Olga Sаukh10 Alexander Ratner1 Shuran Song2 Hannaneh Hajishirzi1,7 Ali Farhadi1 Romain Beaumont6 Sewoong Oh1 Alexandros G. Dimakis5 Jenia Jitsev6,8 Yair Carmon3 Vaishaal Shankar4 Ludwig Schmidt1,6,7

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 Verga
Cade Gordon2 §§ oo Ross Wightman1 §§ Mehdi Cherti1,10
Theo Coombes1 Aarush Katta1 Clayton Mullis1 Mitchell Wodrich
Patrick Schramowski1,4,5 Srivatsa Kundurthy1 Katherine Crowson
Ludwig Schmidt6 oo Robert Kaczmarczyk1,7 oo Jenia Jitsev1,10 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* Wenyu Wang* Dianzhuo Wang
Noah A. Smith♣ Yejin Choi♣ Hannaneh Hajishirzi♣
A bird has four legs.

Plausibility: 15%

2 Knowledge Bases
19 QA datasets
~7M statements

Original example:
What would someone wear to protect themselves from a cannon?
(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)

[Examples adapted from Lin et al., 2020]
Solving Commonsense Benchmarks

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Domain</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>OpenBookQA</td>
<td>scientific</td>
<td>multiple-choice (4)</td>
</tr>
<tr>
<td>ARC (easy)</td>
<td>scientific</td>
<td>multiple-choice (4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Stage B Training (Seen)**

**Evaluation (Unseen Type 1)**

- WSC
- COPA
- NumerSense
- PROST
- Spatial Commonsense

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

**Evaluation (Unseen Type 2)**

- SWAG
- HellaSwag
- CODAH
- Story Cloze Test
- oNLI
- StrategyQA
- CREAK

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

**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)

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!