Making Transformers Curious with Self-Notes

Shubham Toshniwal
NVIDIA

November 10, 2023
NLP Research: What Changed?
NLP Research: What Changed?

LLM

Model Size
Data
FLOPs

LM
Transformer: Backbone of LLMs
Reading or generating $N$ tokens with transformers requires $f(N)$ time.
Reading or generating $N$ tokens with transformers requires $f(N)$ time

Runtime complexity of human language processing also depends on the content and not just the length
Reading or generating $N$ tokens with transformers requires $f(N)$ time.

Runtime complexity of human language processing also depends on the content and not just the length.
One of the main limitations is that the architecture does not allow for an “inner dialogue” or a “scratchpad”, beyond its internal representations, that could enable it to perform multi-step computations or store intermediate results.

— Sparks of Artificial General Intelligence, Bubeck et al. 2023
Transformer: Inner Workings

One of the main limitations is that the architecture **does not allow for an “inner dialogue” or a “scratchpad”**, beyond its internal representations, that could enable it to perform multi-step computations or **store intermediate results**.

— Sparks of Artificial General Intelligence, Bubeck et al. 2023

Lack of Explicit Storage for Intermediate “Thoughts”
been for him. The Witch was turning our troops into stone right and left. But nothing would stop him. He fought his way through three ogres to where she was just turning one of your leopards into a statue. And when he reached her he had sense to bring his sword smashing down on her wand instead of trying to go for her directly and simply getting made a statue himself for his pains. That was the mistake all the rest were making. Once her wand was broken we began to have some chance—if we hadn’t lost so many already. He was terribly wounded. We must go and see him."

They found Edmund in charge of Mrs. Beaver a little way back from the fighting line. He was covered with blood, his mouth was open, and his face a nasty green color.

"Quick, Lucy," said Aslan.

And then, almost for the first time, Lucy remembered the precious cordial that had been given her for Christmas present. Her hands trembled so much that she could hardly undo the stopper, but she managed it in the end and poured a few drops into her brother’s mouth.

"There are other people wounded," said Aslan while she was still looking eagerly into Edmund’s pale face and wondering if the cordial would have any result.

"Yes, I know," said Lucy crossly. "Wait a minute."

"Daughter of Eve," said Aslan in a graver voice, "others also are at the point of death. Must more people die for Edmund?"

"I’m sorry, Aslan," said Lucy, getting up and going with him. And for the next half-hour they were busy—she attending to the wounded while he rescued those who had been turned into stone. When at last she was free to come back to Edmund she found him standing on his feet and not only healed of his wounds but looking better than she had seen him look—oh, for ages; in fact, ever since his first term at that hot school which was where he had begun to go wrong. He had become his real self again and could look you in the face. And there on the field of battle Aslan made him a knight.

"Does he know?" whispered Lucy to Susan.

"What, Aslan? Did he? Does he know what the arrangement with the Witch really was?"

"Hush! No. Of course not," said Susan.

"Oughtn’t he to be told?" said Lucy.

"Oh, surely not," said Susan. "It would be too awful for him. Think how you’d feel if you were he."

"All the same I think he ought to know,"

106
Desiderata

Input/Output processing time should be a function of the content as well

“Memory” for storing intermediate results
Desiderata

Input/Output processing time should be a function of the content as well.

“Memory” for storing intermediate results.

Minimal to no changes to the transformer architecture.
Making Transformers Curious with Self-Notes

Context $\xrightarrow{N_{start}}$ Self-Notes

$Lanchantin^*, Toshniwal^*, Szlam, Weston, Sukhbaatar$

Learning to Reason and Memorize with Self-Notes, NeurIPS 2023
Multi-step Reasoning

The key is in the bag. Alice has the bag.
Multi-step Reasoning

The key is in the bag. Alice has the bag. **Who has the key?**
The key is in the bag. Alice has the bag. **Who has the key?** Alice has the key.
Multi-step Reasoning

Alice has the key. Alice is at the lake.
Multi-step Reasoning

Alice has the key. Alice is at the lake. Where is the key? The key is at the lake.
Multi-step Reasoning

The key is in the bag. Alice has the bag. Alice is at the lake.
Multi-step Reasoning

The key is in the bag. Alice has the bag. Alice is at the lake. **Where is the key? The key is at the lake.**
Multi-step Reasoning: Self-Notes

The key is in the bag. Alice has the bag. Who has the key? Alice has the key. Alice is at the lake.
The key is in the bag. Alice has the bag. **Who has the key? Alice has the key.** Alice is at the lake. **Where is the key? The key is at the lake.**
Tasks

**Toy-story Task**

Original context:
The banana is inside the box. Jessie has the bag. The ball is inside the box. The key is inside the suitcase. Sid has the box. Buzz has the suitcase. Woody is at the station.

Q: Who has the key?

A: Buzz has the key.

**Algorithmic Tasks**

Original context: Context w/ self-Notes:

```
e = 3;
e++; i = 4;
e--; if i > e: e--; g = 3;
Q: print e
```

A: `e = 2`

```
print e e = 3;
e++; print e e = 4;
i = 4; print i i = 4;
e--; print e e = 3;
if i > e: e--; print e e = 2;
g = 3; print g g = 3;
Q: print e
```

A: `e = 2`

**Chess Tasks**

Original context: Context w/ self-Notes:

```
a1 b8 a1 P b8
a1 b1 a1 B b1
a3 c4 a3 P c4
a1 d1 a1 R d1
b6 h7 b6 N h7
b3 c5 b3 K c5
c5 d5 c5 P d5
f5 d5 f5 P d5
e4 f8 e4 K f8
d3 d3 Q
```

Q: print move

A: `e3`

Q: print move

A: `e3`
Supervised Setting

GPT2 model (117 million params) fine-tuned on 10K-500K examples

Access to ground truth self-notes during training

During inference, the model generates its own self-notes
Methods

Vanilla Context Q Model Context Q A
Methods

Vanilla

Context $Q$ → Model → Context $Q$ $A$

Scratchpad or Chain-of-Thought

Context $Q$ → Model → Context $Q$ Reasoning $A$
Methods

**Vanilla**

Context → Q → Model → Context → Q → A

**Scratchpad or Chain-of-Thought**

Context → Q → Model → Context → Q → Reasoning → A

**Self-Notes**

Context → Q → Model → Context → Self-Note → .te. → Self-Note → .xt → Q → Self-Note → A
### Results

<table>
<thead>
<tr>
<th>Task</th>
<th>Test Set</th>
<th>Vanilla</th>
<th>Scratchpad</th>
<th>Self-Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Toy-Story</strong></td>
<td>1/2-hop</td>
<td>92.4 ±0.7</td>
<td>99.6 ±0.1</td>
<td>99.8 ±0.1</td>
</tr>
<tr>
<td></td>
<td>3-hop*</td>
<td>57.0 ±0.3</td>
<td>96.4 ±0.9</td>
<td><strong>98.5 ±0.3</strong></td>
</tr>
<tr>
<td></td>
<td>4-hop*</td>
<td>37.4 ±0.8</td>
<td>94.2 ±2.0</td>
<td><strong>97.8 ±0.4</strong></td>
</tr>
<tr>
<td><strong>Algorithmic</strong></td>
<td>2-100</td>
<td>44.6 ±1.0</td>
<td>72.2 ±5.7</td>
<td><strong>95.5 ±0.2</strong></td>
</tr>
<tr>
<td></td>
<td>101-200*</td>
<td>24.4 ±2.1</td>
<td>11.6 ±2.0</td>
<td><strong>85.0 ±0.6</strong></td>
</tr>
<tr>
<td><strong>Boolean Variable</strong></td>
<td>3-8</td>
<td>99.7 ±0.1</td>
<td>100.0 ±0.0</td>
<td>100.0 ±0.0</td>
</tr>
<tr>
<td></td>
<td>9-19*</td>
<td>71.3 ±0.8</td>
<td>73.7 ±2.4</td>
<td><strong>75.2 ±2.1</strong></td>
</tr>
<tr>
<td><strong>Chess Piece Type</strong></td>
<td>≤80</td>
<td>98.5 ±0.4</td>
<td>98.5 ±0.3</td>
<td><strong>98.8 ±0.2</strong></td>
</tr>
<tr>
<td></td>
<td>≥81*</td>
<td>82.9 ±2.3</td>
<td><strong>94.7 ±0.7</strong></td>
<td><strong>94.8 ±0.7</strong></td>
</tr>
<tr>
<td><strong>Chess Move</strong></td>
<td>≤80</td>
<td>49.0 ±0.4</td>
<td>37.0 ±0.8</td>
<td><strong>50.8 ±1.1</strong></td>
</tr>
<tr>
<td></td>
<td>≥81*</td>
<td>39.8 ±0.2</td>
<td>29.9 ±0.8</td>
<td><strong>41.8 ±0.9</strong></td>
</tr>
</tbody>
</table>
# Results

<table>
<thead>
<tr>
<th>Task</th>
<th>Test Set</th>
<th>Vanilla</th>
<th>Scratchpad</th>
<th>Self-Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Toy-Story</strong></td>
<td>1/2-hop</td>
<td>92.4 ±0.7</td>
<td>99.6 ±0.1</td>
<td>99.8 ±0.1</td>
</tr>
<tr>
<td></td>
<td>3-hop*</td>
<td>57.0 ±0.3</td>
<td>96.4 ±0.9</td>
<td>98.5 ±0.3</td>
</tr>
<tr>
<td></td>
<td>4-hop*</td>
<td>37.4 ±0.8</td>
<td>94.2 ±2.0</td>
<td>97.8 ±0.4</td>
</tr>
<tr>
<td><strong>Algorithmic</strong></td>
<td>2-100</td>
<td>44.6 ±1.0</td>
<td>72.2 ±5.7</td>
<td>95.5 ±0.2</td>
</tr>
<tr>
<td></td>
<td>101-200*</td>
<td>24.4 ±2.1</td>
<td>11.6 ±2.0</td>
<td>85.0 ±0.6</td>
</tr>
<tr>
<td><strong>Boolean Variable</strong></td>
<td>3-8</td>
<td>99.7 ±0.1</td>
<td>100.0 ±0.0</td>
<td>100.0 ±0.0</td>
</tr>
<tr>
<td></td>
<td>9-19*</td>
<td>71.3 ±0.8</td>
<td>73.7 ±2.4</td>
<td>75.2 ±2.1</td>
</tr>
<tr>
<td><strong>Chess Piecetype</strong></td>
<td>≤80</td>
<td>98.5 ±0.4</td>
<td>98.5 ±0.3</td>
<td>98.8 ±0.2</td>
</tr>
<tr>
<td></td>
<td>&gt;81*</td>
<td>82.9 ±2.3</td>
<td>94.7 ±0.7</td>
<td>94.8 ±0.7</td>
</tr>
<tr>
<td><strong>Chess Move</strong></td>
<td>≤80</td>
<td>49.0 ±0.4</td>
<td>37.0 ±0.8</td>
<td>50.8 ±1.1</td>
</tr>
<tr>
<td></td>
<td>&gt;81*</td>
<td>39.8 ±0.2</td>
<td>29.9 ±0.8</td>
<td>41.8 ±0.9</td>
</tr>
</tbody>
</table>
Vanilla

```python
e = 3 ;
e ++ ;
i = 3 ;
if i < e : e ++;
print e
```

```python
e = 5 ;
```

Scratchpad

```python
e = 3 ;
e ++ ;
i = 3 ;
if i < e : e ++;
print e
```

```python
[e = 3 ;
e ++ ;
print e e = 4 ;
i = 3 ;
if i < e : e ++ ;
print e e = 5 ;]
e = 5 ;
```

Self-Notes

```python
e = 3 ;
e ++ ;
print e e = 4 ;
i = 3 ;
if i < e : e ++;
print e e = 5 ;
```

```python
e = 5 ;
```

Self-Notes avoids copying the input
Reduced Supervision Setting

Previous experiments assumed access to 100% of training instances labeled with intermediate reasoning steps.

What happens if we lack annotations:

- *Semi-supervised*: Part of the training data labeled with Self-Notes
- *Unsupervised*: None of the training data labeled with Self-Notes
Semi-Supervised Self-Notes

Treat it as a multitask learning problem

No Self-Notes

Self-Notes
Toy-Story: Just 5% of Self-Notes labeled data needed to significantly outperform the baseline model.
Unsupervised Self-Notes

**Assumption:** The final task and the intermediate task have similar input-output pairs

**Solution:** Enrich training data with model generated Self-Notes

- Generate Self-Notes (QA pairs/print statements) for partial inputs in the training data
- Filter *high-confidence* Self-Notes
- Finetune the model on filtered model generated Self-Notes
Unsupervised Self-Notes: Toy Story

![Bar chart showing accuracy for Toy Story tasks with and without self-notes. The chart compares vanilla and self-notes (unsupervised) for 3-hop and 4-hop tasks. The accuracy is measured in percentage.]
Can Self-Notes work with just a few labeled examples?
Can Self-Notes work with just a few labeled examples?

Test Self-Notes in the few-shot prompting/in-context learning paradigm with LLAMA2 (70B)

Input-1 Output-1 Input-2 Output-2 · · · Input-k Output-k Test Input
LLMs: Few-shot Prompting

Arithmetic Word Problems: MultiArith and GSM8K

Q: There are 15 trees in the grove. (15 total) Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. (21 - 15 = 6 left) How many trees did the grove workers plant today?
A: The answer is 6.

Q: Leah had 32 chocolates and her sister had 42. (32 + 42 = 74 total) If they ate 35, (74 - 35 = 39 left) how many pieces do they have left in total?
A: The answer is 39.

Q: Shawn has five toys. (5 total) For Christmas, he got two toys each from his mom and dad. (2 + 2 = 4 more) How many toys does he have now? (5 + 4 = 9 total)
A: The answer is 9.

Few-shot examples from MultiArith
Results: Few-shot Prompting

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Vanilla</th>
<th>CoT</th>
<th>Self-Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithmic</td>
<td>23.3</td>
<td>37.4</td>
<td>40.3</td>
</tr>
<tr>
<td>MultiArith</td>
<td>33.3</td>
<td>94.5</td>
<td>96.2</td>
</tr>
<tr>
<td>GSM8K</td>
<td>16.5</td>
<td>55.9</td>
<td>59.9</td>
</tr>
</tbody>
</table>
Takeaways

Across multiple tasks and multiple learning paradigms, Self-Notes outperforms Vanilla/Scratchpad/Chain-of-Thought baselines
Takeaways

Across multiple tasks and multiple learning paradigms, Self-Notes outperforms Vanilla/Scratchpad/Chain-of-Thought baselines

Strengths
- Aids the alignment of reasoning steps to partial input
- Extends the working memory
- Debugging mistakes can be easier than generating the entire reasoning trace at the end (Chain-of-Thought)

Future Work
- Long-context QA tasks
- Code generation with program state
Takeaways

Across multiple tasks and multiple learning paradigms, Self-Notes outperforms Vanilla/Scratchpad/Chain-of-Thought baselines

Strengths
- Aids the alignment of reasoning steps to partial input
- Extends the working memory
- Debugging mistakes can be easier than generating the entire reasoning trace at the end (Chain-of-Thought)

Future Work
- Long-context QA tasks
- Code generation with program state