TALM and the state of Tool-Use

Aaron Parisi, Google Deepmind

Coauthors: Noah Fiedel, Yao Zhao

Agenda

1. Why tool-use
2. TALM overview
3. Lessons learned
4. The future of tool-use

Why tool-use?

Lots of obvious reasons

- **Modularity**
- **Offloads computations / functions**
- **Parameter efficiency** - fewer computations, simpler computations -> fewer parameters needed

Oh yeah and I do suppose

- It's **one of the hallmarks of sapient intelligence on Earth**
- **Tools** are essential for **addressing the shortcomings of any "thinking system"**

The progress of tool-use

- **Markov Decision Processes** (MDPs) expose discrete actions
  - **Heirarchical MDPs** allow for parametric tool use
- Custom models (**fuzzy logic, RL agents, domain specific languages**) for parametric tool-use
- **Language models** expand sequence modelling (MDP representation) capabilities
- **TALM goes here**
- **Few-shot / 0-shot prompting**
- Language models **begin composing** more robust API calls

TALM

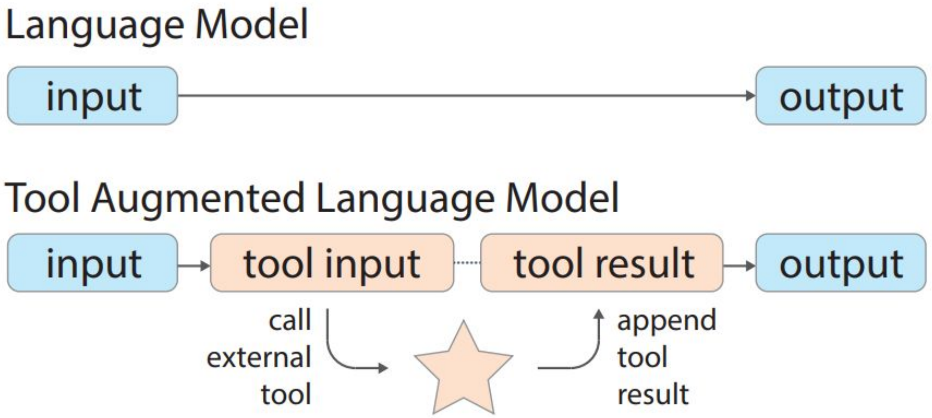
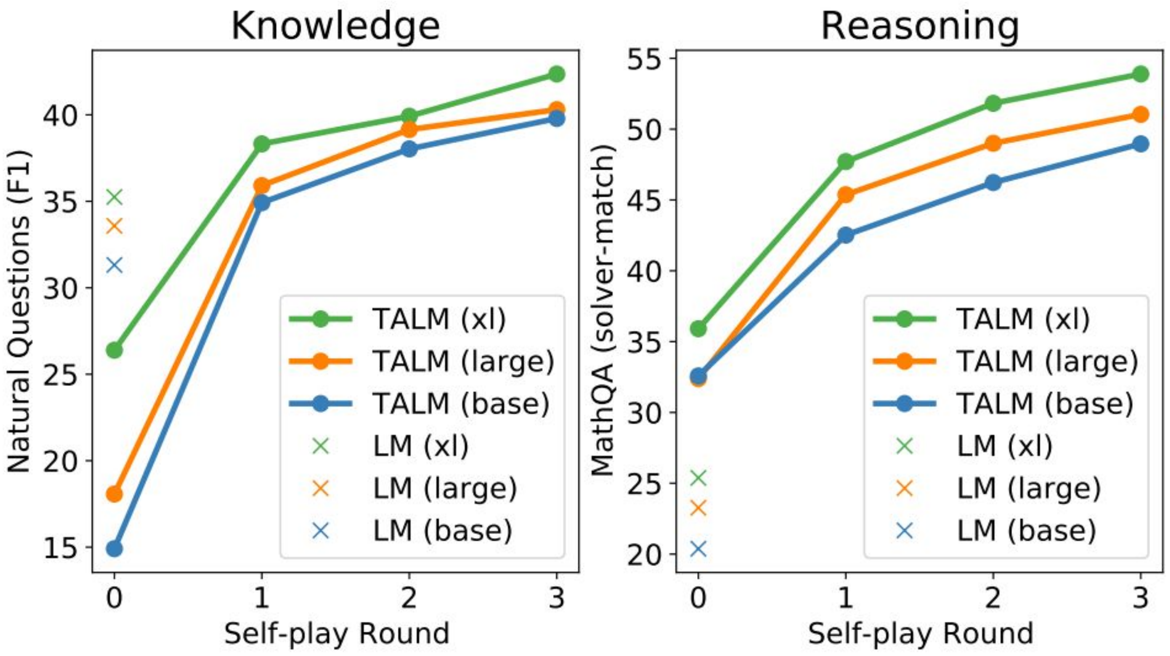


Figure 2: LM and Tool Augmented LMs.

Results overview



Algorithm Overview

- **Expert Iteration**
- **Obvious Drawbacks**: Search space increases exponentially as the induced MDP expands (more tools / steps -> exponentially growing search space)
    - REINFORCE with binary reward signal (good vs bad outcome) is a heavily biased estimator

Algorithm 1 Iterative Self-Play Algorithm.

x: task input, y: task output, t: tool input, r: tool output

$T = \{x_i, y_i\}_T$ # task set

$D = \{x_j, t_j, r_j, y_j\}_D$ # tool-use set

$P_\theta \leftarrow pretrained\ LM$

for $t \in [0, 1, ..., R]$ **do** # self-play rounds

finetune LM

$\quad \theta \leftarrow \underset{\theta}{\text{argmax}} \prod_D P_\theta(y_j|x_j, t_j, r_j) P_\theta(t_j|x_j)$

$\quad$ **for** $x_i, y_i \in T$ **do** # iterate task set

$\quad\quad$ **for** $n \in [0, 1, ..., N]$ **do**

$\quad\quad\quad t_n \leftarrow P_\theta(t|x_i)$ # sample tool query

$\quad\quad\quad r_n \leftarrow Tool(t_n)$ # call tool API

$\quad\quad\quad y_n \leftarrow P_\theta(y|x_i, t_n, r_n)$ # get task output

$\quad\quad\quad$ **if** $|y_n - y_i| < th$ **then** # filter wrong output

$\quad\quad\quad\quad D \leftarrow D \cup \{x_i, t_n, r_n, y_n\}_1$

update tool-use set

Lessons Learned

A bitter lesson

- Shortly after publishing TALM, instruct-tuned models really became a thing
- Why go through such an exhaustive search procedure when few-shot/0-shot/prompt tuning methods work?
  - This method **only really makes sense** for smaller models that don't benefit from prompting methods.
- **Furthermore**, we found that larger language models need significantly fewer finetuning examples to be able to learn tool-use without few-shot examples!

A useful tech demo

- **Bootstrapping works!**
- **Toolformer** - obvious extension to our work
  - Let's augment the loss function with a reward function that tries to signal the **causal effects of tool-use** at a given timestep
  - This is similar to augmenting the binary self-play reward signal to be less biased towards cases where the model would succeed without tool-use
- Smaller models can **reliably bootstrap their own performance** for simple tasks, dependent almost entirely on the performance of the search procedure

The state, and future of tool-use

Tool-Use and Large Models

- Few-shot/zero-shot/prompt-tuning for **large models**
  - **Open Questions:**
    - How do we get LLMs to handle arbitrary, increasingly complex tools?

Tool-Use and Small Models

- **Data augmentation via self-play for small models?**
    - **Open Questions:**
        - How do we prevent the search space from growing exponentially?
            - **More robust data augmentation, representations?**
        - Can small models compose like big models?

Conclusion and Q&A

Both large and small models benefit from sampling/training algorithmic improvements! Stay tuned for some exciting advancements from GDM, of course!

[specifically, look forward to a paper addressing the shortcomings of outcome-based (binary reward) RL on implicit MDPs]

[**and maybe get my personal contact info too:**

aarontp@proton.me or @ me on linkedin, I have personal ambitions ;P]