Learning to Reason and Memorize with Self-Questioning

Jack Lanchantin, Shubham Toshniwal, Jason Weston, Arthur Szlam, Sainbayar Sukhbaatar

∞ **Meta AI**

Overview

- Large language models have been shown to struggle with limited context memory and multi-step reasoning
- We propose a method for solving both of these problems by allowing the model to ask and answer questions
- The model can deviate from the input context at any time for self-questioning
- This allows the model to recall information and reason on the fly as it reads context, extending its memory and enabling multi-step reasoning

Multi-hop reasoning

Given "**Alice has the box.**" and "**Alice is at the park.**" one can infer "**The box is at the park.**", which can be further combined with a later statement "**The key is in the box.**" to conclude that "**The key is at the park.**".

State-tracking

Assume **x=5** initially, and then **x** gets incremented by 1. If the model correctly writes **x=6**, it can safely remove the original **x=5** statement from its context.
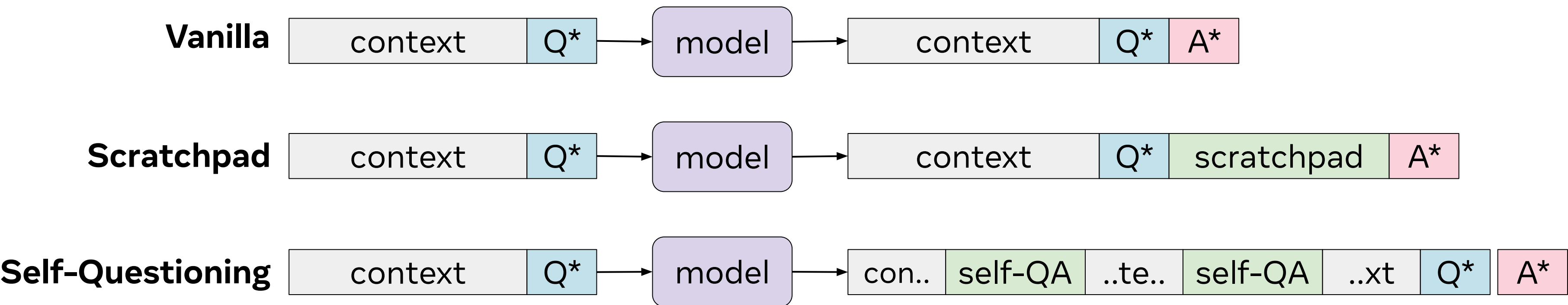
Method: Self-questioning

2 hop question:

Bob has the key. Bob is at the park.
Q: where is the key? The key is at the park

3 hop question:

The ball is in the bag. Alice has the bag. Alice is at the lake.
Q: where is the ball? The ball is at the lake

3 hop question with self-questioning:

The ball is in the bag. Alice has the bag. SQ: who has the ball? Alice has the ball. Alice is at the lake.
Q: where is the ball? The ball is at the lake

Model

Data

Green tokens are the self-QAs, Red tokens are the final answer. "print" and "SQ:" are special tokens to generate a self-QA for the Algorithmic and Toy-story, respectively. Ithe model predicts the special token as the next token, it is allowed to generate a question and answer before returning to the original context.

Results

Method	Train self-QA	Test self-QA	Algorithmic			Toy-Story	
			≤50	≤100	≤200*	3-hop*	4-hop*
GPT-2 fine-tuned	none	none	69.5	53.2	40.0	59.8	39.4
Scratchpad	100%	generated	**100.0**	61.0	30.0	98.6	95.4
Self-Questioning	100%	generated	**100.0**	**100.0**	**98.8**	**99.7**	**98.6**
	75%		99.5	99.6	98.3	98.9	98.5
	50%		98.8	98.2	95.0	98.9	96.8
	25%		95.8	93.5	85.7	97.1	95.1
Ablation	100%	100%	100.0	100.0	100.0	100.0	100.0
	100%	none	25.0	16.3	15.9	39.7	32.5

Test Accuracy (in %). Train and test columns show what percentage of samples in the split have access to ground truth self-QAs, or if they are generated by the model itself. "*" indicates out-of-distribution harder test settings.

Example output

tl;dr - a general method that allows language models to take internal notes in the form of self-QAs

Paper link: