Can LLMs solve generative visual analogies?

Shrey Pandit[1], Gautam Shroff[2], Ashwin Srinivasan[1] and Lovekesh Vig[2]

[1]BITS Pilani, K.K. Birla Goa Campus, India
[2]TCS Research,New Delhi, India

Abstract

Recent experiments with large language models (LLMs) have provided some evidence that these models can perform abstract analogical reasoning [1], including textual puzzles similar to Raven's progressive matrices. We consider a visual analogical reasoning task that was solved using neuro-symbolic techniques in [2], and investigate how LLMs fare on this task. The task involves learning a sequence of transformations by which a sample input/output pair of images are related so as to analogously transform a test input. Note that unlike the analogical reasoning tasks in [1], this task involves *generating* an output as opposed to selecting from a set of choices. We evaluated various LLMs including GPT-4, GPT 3.5-turbo (ChatGPT), and GPT3 on this task for differing lengths of the sequence of transformations relating the input and output. Our results suggest that GPT-4 performs the best overall, while GPT 3.5-turbo and GPT3 perform strongly on shorter program lengths. At the same time, the performance of LLMs for this task falls far short of the neuro-symbolic approach used earlier, and we speculate as to why this may be the case, at least as of now.

Keywords

Large language models, GPT-4, Visual analogy, Neural analogical reasoning

1. Introduction

As in [2] we consider the class of visual reasoning problems as demonstrated in Figure 1c in which each task involves a *functional analogy* (i.e., $x : f(x) :: y : f(y)$) wherein each shape in the input image (here just one) is transformed to one or more positions in the output image via a sequence of elementary transformations, e.g., shifts in the 3x3 grid. Given a solved example, constructing the analogous output for a test input can be cast as a program synthesis problem where we seek to discover a program consisting of one or more sequences of shifts that need to be applied to each input shape in order to generate the output image. This program can then be applied to a text image to generate an *analogous* output.

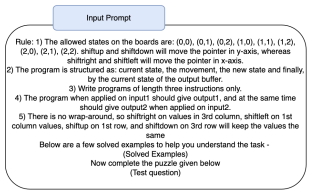
The neuro-symbolic approach in [2] achieved 100% success on a large collection of such problems when presented in symbolic form and 94.7% success when given images. In contrast, pure deep-learning approaches, including meta-learning could achieve 74% success at best as also documented in [2].

IARML@IJCAI'2023: Workshop on the Interactions between Analogical Reasoning and Machine Learning, at IJCAI'2023, August, 2023, Macao, China
*Corresponding author.
pandit.shrey.01@gmail.com (S. Pandit)

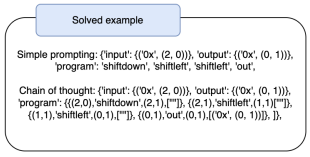
2. Experiments

We applied LLMs to solve such a visual analogy task, with the image translated into symbolic form. We use the trained models provided by OpenAI API and give a few solved examples in the prompt to help the LLMs learn.

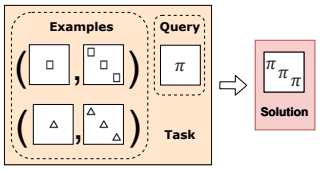
Prompting We prompt the LLMs with a set of rules for the task, which includes information on the allowed positional shifts, the set of permissible states, and the non-wrapping state of the grid. As a hint, we also specify the expected program length in the prompt. We also provide a set of solved examples to guide the LM's learning process. Figure 1a illustrates a representative example of the prompt.



(a) Prompt with rules, solved examples and test input.



(b) Chain-of-thought vs. simple prompting



(c) Given example input-output image pairs, generate the analogous output for the given query image.

Figure 1: Figure 1a showing the prompt rules, Figure 1b showing the comparison of COT vs. simple prompting, and the Figure 1c showing overview of the entire process.

		Program length 3		Program length 5	
	Best of n outputs	Simple prompting	Chain of thoughts	Simple prompting	Chain of thoughts
GPT-3	1	14%	20%	5.4%	4.16%
	3	26%	30%	11.1%	12.8%
	5	30%	42%	23%	18.9%
GPT-3.5-turbo	1	10%	6%	6.12%	8%
	3	16.3%	14%	22%	18%
	5	33.3%	22%	12%	14%
GPT-4	1	18.36%	26%	22%	12%
	3	38.77%	52%	26.5%	22%
	5	40%	46%	39.58%	34%

Table 1
Table comparing the performance of GPT3, GPT 3.5-turbo, and GPT4 over program lengths 3 and 5, with various different numbers of outputs and ways of prompting over 50 trials. In the above case, we are providing ten solved examples.

Sampling The top prediction generated by a LM may not always be the optimal choice, so we also evaluate results using the top-n predictions for consideration in determining the correctness of the task. The task is deemed successful if *any* of the n predictions are accurate.

GPT3			
Simple Prompting	Tokens	2847	3690
	Accuracy	48%	32%
Chain-of-thought prompting	Tokens	2345	3813
	Accuracy	42%	50%

Table 2
Chain-of-thought vs more examples for same token length.

Chain-of-thought prompting: Previous works such as [3] have shown that language-model performance increases drastically in reasoning tasks when given chain-of-thought prompts. In our context we provide a chain-of-thought by providing, with each step of the solved example, the current state on the grid, positional shift, next position on grid, and the state of the output buffer.
Changing the program length: We experimented with different program lengths (3 & 5); empirically, increasing the program length makes the task more difficult.

3. Results and Conclusions

Referring to Table 1 we observe the following: (i) Chain-of-thought improves performance over simple prompting, which was expected. (ii) Further, chain-of-thought prompting is also better (albeit slightly) than providing more examples for similar token lengths, see Table 2. (ii) Sampling more examples improves performance, also expected. (iii) Analogies involving longer sequences (programs) are more difficult as expected; however we observe a drastic drop in performance for GPT3 and GPT 3.5-turbo but only a marginal drop is observed for GPT4. While the input prompts do affect the LLMs' performance, it's important to mention that a uniform prompt template was used for all the analyzed LLMs in this study.

Overall the performance of LLMs for our simple visual analogy task fall far short of the neuro-symbolic techniques used in [2]. We note that [2] relied *search* over possible sequences that could successfully transform a text input to its output. LLMs do not explicitly search over potential outputs. We speculate that incorporating elements of explicit search may enable LLMs to perform better at generative analogies.

References

[1] W. et. al, Emergent analogical reasoning in large language models, arXiv:2212.09196 (2022).
[2] S. et. al, Solving visual analogies using neural algorithmic reasoning, AAAI (Student Abstract) (2022).
[3] J. W. et. al, Chain of thought prompting elicits reasoning in large language models, NeurIPS 2022 (2022).