PNAS NEXUS

PNAS Nexus, 2025, 4, pgaf135
https://doi.org/10.1093/pnasnexus/pgaf135
Advance access publication 27 May 2025
Brief Report

Evidence from counterfactual tasks supports emergent analogical reasoning in large language models

Taylor W. Webb [a,*], Keith J. Holyoak [b] and Hongjing Lu [b,c]

[a]Microsoft Research, New York, NY 10012, USA
[b]Department of Psychology, University of California, Los Angeles, CA 90095, USA
[c]Department of Statistics, University of California, Los Angeles, CA 90095, USA
*To whom correspondence should be addressed: Email: taylor.w.webb@gmail.com
Edited By Mohammad Atari

Abstract

A major debate has recently arisen concerning whether large language models (LLMs) have developed an emergent capacity for analogical reasoning. While some recent work has highlighted the strong zero-shot performance of these systems on a range of text-based analogy tasks, often rivaling human performance, other work has challenged these conclusions, citing evidence from so-called "counterfactual" tasks—tasks that are modified so as to decrease similarity with materials that may have been present in the language models' training data. Here, we report evidence that language models are also capable of generalizing to these new counterfactual task variants when they are augmented with the ability to write and execute code. The results further corroborate the emergence of a capacity for analogical reasoning in LLMs and argue against claims that this capacity depends on simple mimicry of the training data.

Keywords: large language, analogical reasoning, counterfactual tasks, zero-shot reasoning, code execution models

The advent of large language models (LLMs) has already had a major influence on the cognitive sciences, with much recent work investigating the potential cognitive capacities of these systems (1), but a major ongoing debate concerns the question of whether these capacities extend to higher cognitive processes such as reasoning (2). One area of particular interest has been the extent to which LLMs possess an emergent capacity for *analogical* reasoning, a central aspect of human intelligence that supports generalization to novel problems and environments (3). Recent work has found that some state-of-the-art LLMs display strong zero-shot (i.e. without task-specific training) performance on a range of text-based synthetic and real-world analogy problems, often performing at or above the level of human performance (4).

However, it has also been argued that these results do not reflect a genuine capacity for analogical reasoning but instead reflect "approximate retrieval," mimicry of similar materials present in the language models' training data. Proponents of this argument have pointed to evidence from "counterfactual" tasks—a term that was recently introduced to refer to unusual task variants, designed to deviate from common problems on which language models may have been trained (5). In particular, two recent studies have highlighted results from a specific variant of the letter-string analogy task, involving a permuted alphabet (see example in Fig. 1a), on which the Generative Pre-trained Transformer (GPT) class of language models (e.g. GPT-3 and GPT-4) display degraded performance (6, 7).

Based on these results, it has been argued that the ability of language models to solve other types of analogy problems is based only on the similarity of those problems to the training data. This conclusion, however, ignores alternative potential explanations for the poor performance of these models on this particular task variant. Most notably, such problems require that letters be converted into the corresponding indices in the permuted alphabet, a process that depends on the ability to precisely count the items in a list. It is well known that language models have difficulty with counting, a phenomenon that may in fact be related to the capacity-limited nature of rapid numerical estimation in visual displays by humans (i.e. "subitizing") (8). Thus, one potential alternative interpretation of these results is that they reflect not a general inability to perform analogical reasoning, but simply a specific difficulty with problems that require counting.

To test this alternative interpretation, we evaluated a more recently released variant of GPT-4, augmented with the capacity to write and execute code. This code execution capacity is often invoked by the model when performing tasks that require precise indexing within a list. Consistent with this alternative interpretation, GPT-4 was able to solve these "counterfactual" letter-string analogies at a roughly human level of performance when given the ability to count using code execution (Fig. 1b; logistic regression, human participants vs. GPT-4 + code execution: $P = 0.92$), whereas without this functionality GPT-4 performed significantly worse, on par with previous results (6, 7) (human participants vs. GPT-4: $P < 2 \times 10^{-16}$).

Competing Interest: Taylor Webb is a postdoctoral researcher at Microsoft Research.
Received: December 13, 2024. **Accepted:** April 4, 2025

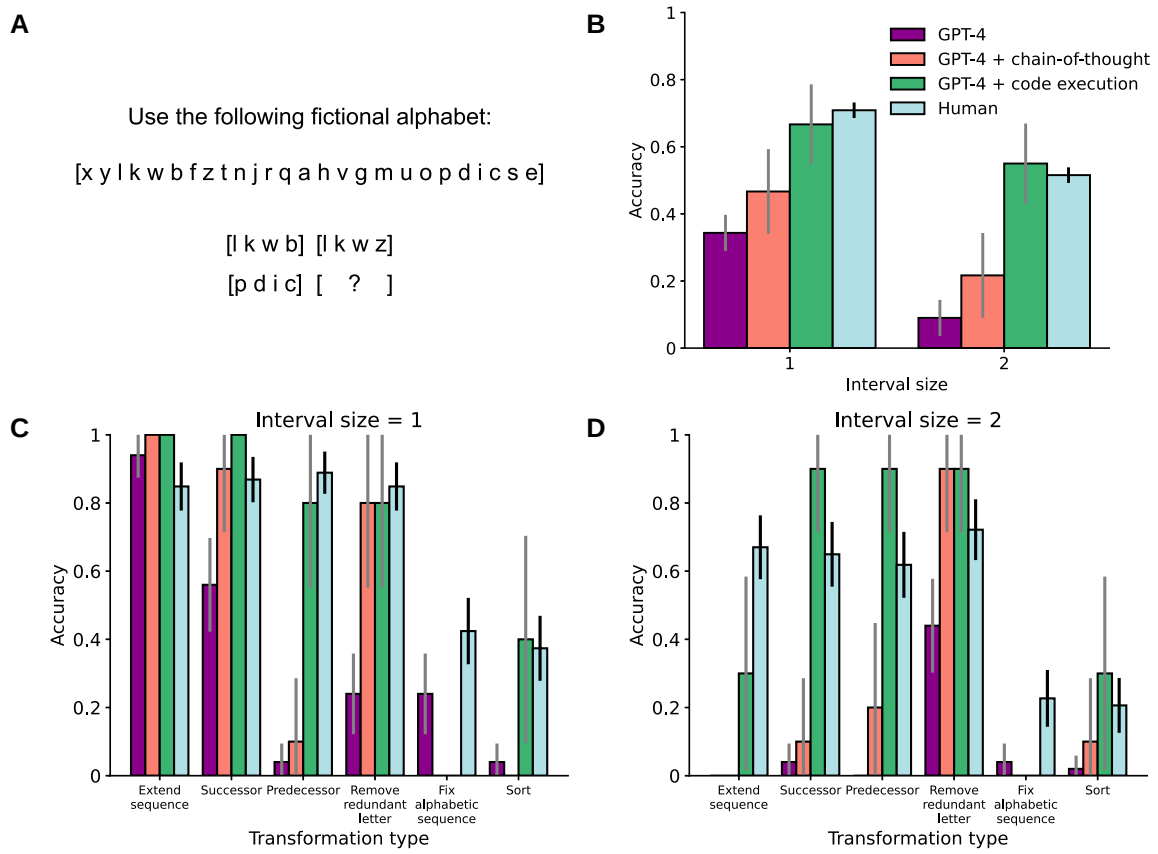

Fig. 1. Results for letter-string analogies with shuffled alphabet. A) Example of a counterfactual letter-string analogy problem constructed using a permuted alphabet. This example involves a successor transformation with an interval size of 1, applied to the final letter of the string. Other problems involved an interval size of 2, and different types of transformations. B) Summary of results for human participants, GPT-4, GPT-4 with zero-shot chain-of-thought, and a variant of GPT-4 augmented with the capacity to write and execute code. C) Results for problems with an interval size of 1, sorted by transformation type. D) Results for problems with an interval size of 2. Human results reflect average performance for $N = 99$ participants for interval-size-1 and $N = 97$ separate participants for interval-size-2. Human behavioral data were collected in an online experiment. The experiment was approved by the UCLA Institutional Review Board, and all participants provided informed consent. Black error bars represent standard error of the mean across participants. Gray error bars represent 95% binomial CIs for average model performance across multiple problems.

We also tested whether zero-shot chain-of-thought (i.e. allowing the model to "think" before producing a final answer) would improve GPT-4's performance, as this approach has been shown to improve performance on precise quantitative tasks like counting (9). We found that, although this approach did improve performance to some extent (GPT-4 vs. GPT-4 + chain-of-thought: $P = 0.004$), GPT-4 with chain-of-thought still performed worse than human participants (human participants vs. GPT-4 + chain-of-thought: $P = 3 \times 10^{-8}$) and GPT-4 with code execution (GPT-4 + code execution vs. GPT-4 + chain-of-thought: $P = 4.4 \times 10^{-5}$). These results suggest that chain-of-thought is not as reliable as code execution for carrying out the indexing operations in these problems. Consistent with this hypothesis, we found that GPT-4 could not reliably identify the interval between two letters in our synthetic alphabet (overall accuracy of 10.8% for identification of intervals between −2 and 2), further suggesting that counting is the source of GPT-4's difficulty on this task.

Importantly, GPT-4 only relied on code execution to convert letters into their corresponding indices, using code that GPT-4 generated on its own. Moreover, it was not necessary to instruct GPT-4 to use code execution in this manner, nor to provide any task-specific instructions (e.g. regarding the importance of position or interval size). In a control experiment, in which the code execution model was instructed *not* to invoke its code execution facility, performance was on par with GPT-4 alone (GPT-4, accuracy = 21.7%; code execution control model, accuracy = 22.5%; logistic regression, GPT-4 vs. control model: $P = 0.73$). This result indicates that improved performance was causally related to the use of code execution, and not to any auxiliary prompts that may be employed in the code execution model.

Looking more closely at the responses generated by the model, we found that correct responses were typically accompanied by a coherent and accurate explanation (see example in Supplementary Section S11). In addition, many incorrect responses were based on a less abstract but nevertheless valid rule, at a rate similar to that observed for human participants (38% of GPT-4's errors involved a valid alternative rule, compared with 39% of errors in the human behavioral results reported in previous work (7)). Of course, the use of code execution to solve these particular problems is not a human-like solution (humans do not need to write code to count the items in a list). But these results suggest that the inability of language models to solve these particular reasoning problems is most likely the result of a specific difficulty with counting, rather than a general inability to solve analogy problems. Given the deliberate "counterfactual" design of this task, GPT-4's ability to solve these problems at a roughly human level, and to provide accurate explanations of its solutions, is difficult to explain in terms of the presence of these problems in the training data.

It might be argued that by converting letters from the permuted alphabet into numerical indices, the code-augmented variant of

GPT-4 effectively rerepresented the counterfactual letter-string problems as more familiar numerical problems (i.e. the counterfactual problems were transformed into a known format). But as early Gestalt psychologists emphasized, human reasoners also make use of rerepresentation to solve insight problems (10). The basic role of analogy in problem solving is to understand an unfamiliar target problem by mapping it onto a more familiar source (11). This aspect of the code-augmented model's approach is thus arguably human-like. It should be emphasized that the code-augmented model autonomously generated code to convert problems into a suitable format—the system as a whole was presented only with the counterfactual problems, and the ability to solve them thus cannot be explained by dependence on similarity to the training data. Importantly, these findings pertain specifically to the domain of analogical reasoning, and do not rule out the possibility that LLMs may depend on "approximate retrieval" in other domains of reasoning (e.g. logical reasoning (5)).

Our results should not be interpreted as suggesting that code execution is generally required for language models to solve counterfactual analogy problems. Indeed, previous work (4), found that language models can solve some types of counterfactual problems, including a variant of letter-string analogies involving generalization from letters to real-world concepts (e.g. p q r : p q s :: cold cool warm : ?), and a novel matrix reasoning task designed specifically to test language models. Instead, in our experiments code execution is employed to assist the language model with a specific auxiliary task (counting) that must be performed to generate the inputs to analogical reasoning.

More generally, these results illustrate an important point about the evaluation of cognitive capacities in artificial systems. A central lesson of cognitive science is that cognition is comprised of interacting, but dissociable, processes. There is no particular reason to expect that these processes will similarly covary in artificial systems, especially those with radically different developmental origins from our own. Thus, as has previously been argued (4), it is important to distinguish domain-specific failures in processes such as physical reasoning—or, in this case, counting—from the evaluation of core competencies such as analogical reasoning. Indeed, human reasoners also display dramatic variability in their capacity to deploy analogical reasoning in specific domains, based on their expertise in those domains (12). This observation echoes recently articulated concerns about the potential confounding influence of auxiliary task demands when evaluating LLMs (13). Just as when testing young children or nonhuman animals, it is important to design evaluations that probe the capacity of interest while avoiding confounds resulting from auxiliary task demands.

A full account of emergent analogical reasoning in LLMs, and of its relationship to human reasoning, will require us to go beyond behavioral evaluations by examining the internal mechanisms that support this capacity. While there is still much to learn, a growing body of evidence suggests that this capacity may be supported by a set of structured operations and emergent relational representations (14). Indeed, in-context learning—the capacity of LLMs to rapidly learn new tasks by conditioning their predictions on a small set of in-context task examples—may itself depend on analogical reasoning (or more generally, schema induction). This possibility is suggested by the dependence of in-context learning on the emergence of structured mechanisms for similarity-based inductive inference (15). Thus, the core mechanisms of few-shot learning and inference in transformer-based language models may depend on reasoning over patterns of similarity, mirroring the central role of similarity in human reasoning (3). It remains an important priority for future work to determine whether and how the mechanisms that support this capacity in LLMs relate to those that implement reasoning in the human brain.

Supplementary Material

Supplementary material is available at *PNAS Nexus* online.

Funding

This research has been supported by an award from Microsoft Azure as part of the Accelerating Foundation Models Research initiative.

Preprints

This manuscript was posted as a preprint available at https://arxiv.org/abs/2404.13070.

References

1 Binz M, Schulz E. 2023. Using cognitive psychology to understand GPT-3. *Proc Natl Acad Sci U S A*. 120(6):e2218523120.
2 Mahowald K, *et al.* 2024. Dissociating language and thought in large language models. *Trends Cogn Sci*. 28(6):517–540.
3 Holyoak KJ. 2012. Analogy and relational reasoning. In: *The Oxford handbook of thinking and reasoning*. Oxford University Press. p. 234–259.
4 Webb T, Holyoak KJ, Lu H. 2023. Emergent analogical reasoning in large language models. *Nat Hum Behav*. 7(9):1526–1541.
5 Wu Z, *et al.* 2024. Reasoning or reciting? Exploring the capabilities and limitations of language models through counterfactual tasks. In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.
6 Hodel D, West J. 2023. Response: Emergent analogical reasoning in large language models. *arXiv*, arXiv:2308.16118. https://doi.org/10.48550/arXiv.2308.16118, preprint: not peer reviewed.
7 Lewis M, Mitchell M. 2024. Using counterfactual tasks to evaluate the generality of analogical reasoning in large language models. In: *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 46).
8 Kaufman EL, Lord MW, Reese TW, Volkmann J. 1949. The discrimination of visual number. *Am J Psychol*. 62(4):498–525.
9 Wei J, *et al.* 2022. Chain-of-thought prompting elicits reasoning in large language models. *Adv Neural Inf Process Syst*. 35:24824–24837.
10 Duncker K. 1945. On problem solving. *Psychol Monogr*. 58(5):1–113.
11 Gick ML, Holyoak KJ. 1980. Analogical problem solving. *Cogn Psychol*. 12(3):306–355.
12 Goldwater MB, Gentner D, LaDue ND, Libarkin JC. 2021. Analogy generation in science experts and novices. *Cogn Sci*. 45(9):e13036.
13 Hu J, Frank MC. 2024. Auxiliary task demands mask the capabilities of smaller language models. In: *First Conference on Language Modeling*.
14 Todd E, *et al.* 2024. Function vectors in large language models. In: *12th International Conference on Learning Representations*.
15 Olsson C, *et al.* 2022. In-context learning and induction heads. *arXiv*, arXiv:2209.11895. https://doi.org/10.48550/arXiv.2209.11895, preprint: not peer reviewed.