On Using Generative Models in a Cognitive Architecture for Embodied Agents

Dongkyu Choi
Institute of High Performance Computing (IHPC)
Agency for Science, Technology and Research (A*STAR)
1 Fusionopolis Way, #16-16 Connexis
Singapore 138632, Republic of Singapore
choi_dongkyu@ihpc.a-star.edu.sg

Abstract

Recent popularity of generative models brought research on a variety of applications. We take a more architectural point of view, where we discuss ways in which generative AI techniques and cognitive architectures can benefit each other for a more capable overall integrated system. We use a cognitive architecture, ICARUS, as the framework for our discussion, but most of the discussed points should carry over to other architectures as well.

Introduction

Recent introduction of ChatGPT[1] brought a flurry of both enthusiasm and skepticism in the field of Artificial Intelligence (AI) and our society in general, attracting exploding interests on Large Language Models (LLMs) and other generative AI techniques. Since then, people have reported on numerous ways of using these tools in a variety of domains. There are straightforward applications of these models as they were originally intended, namely, as language models to understand and generate natural language sentences. Examples include utterance generation from keywords, rewriting input sentences, summarizing given paragraphs, translation from one language to another, and so forth.

But there are other proposed applications where language serves as an intermediate medium for a different goal. In such cases, researchers use LLMs as a common-sense knowledge base, not simply as a language model. Some examples include arithmetic calculations (Chen et al. 2022) and logical reasoning (Wei et al. 2022; Kojima et al. 2022; Talmor et al. 2020). Many of these previous work involve using LLMs in zero-shot or few-shot settings, namely, using them in their pre-trained or fine-tuned state without any further training. This requires what is known as *prompt engineering* that deals with various ways one can provide information to LLMs and elicit appropriate responses from them, sometimes using examples from the application domain.

Yet another type of work uses generative models in an embodied setting, especially for the purpose of task planning. Wang et al. (2023a) presents a system that is capable of long-horizon exploration in an open-ended game using multiple instances of LLMs through prompting. Other cases employ further training of LLMs beyond their pre-trained state using domain-specific knowledge. For instance, Driess et al. (2023) uses an end-to-end trained LLM that takes multimodal input and returns high-level actions in text.

In this paper, we take a more architectural point of view, where we explore and identify potential uses for LLMs and other generative models in the context of cognitive architectures. Although we are using ICARUS (Langley and Choi 2006; Choi and Langley 2018) as the framework of choice, most, if not all, of the discussed points should also be applicable to other cognitive architectures. Our discussion will not be specific to the unique features of ICARUS and stay more focused on common architectural ones. In the next sections, we will first review the cognitive architecture briefly, highlighting its theoretical commitments that are shared with other architectures. Then we will identify where in its cognitive processes the architecture can benefit from generative models' capabilities and discuss how using these models in an architectural context can help remedy one of their important drawbacks.

Review of ICARUS Cognitive Architecture

ICARUS is a cognitive architecture that provides a computational framework for modeling general intelligence. It makes a specific set of commitments as to how to represent knowledge, where to store it, and how to process it in what kind of cognitive machinery. The architecture focuses on embodied agents in physical domains by grounding its inference on perceived objects and its execution on actions in the world. ICARUS shares some of its commitments with other architectures like Soar (Laird, Newell, and Rosenbloom 1987; Laird 2012) and ACT-R (Anderson and Lebiere 1998). Some commonalities include the separation of long-term knowledge and short-term structures, the distinction between conceptual and procedural knowledge, and its operation in recognize-act cycles. There is, however, a distinct combination of features unique to ICARUS. For instance, the architecture features goal-driven, but reactive execution with top-level goal reasoning capabilities, using hierarchically organized concepts and skills.

Figure 1 shows the organization of ICARUS's memories and the processes that operate over them. On each cycle, the architecture receives from the environment a list of per-

[1] https://openai.com/chatgpt

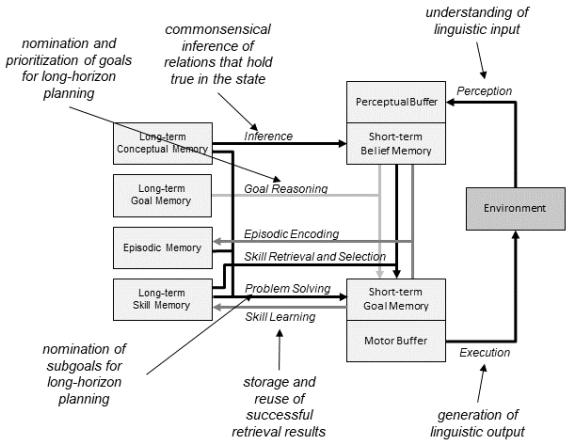

Figure 1: A block diagram showing the ICARUS cognitive architecture's memories and buffers and the processes that operate over them. The notes around the block diagram give potential use of generative AI in the architectural processes.

ceived objects along with their attributes in its perceptual buffer. Based on this information, ICARUS performs pattern matching to infer all the instances of its concepts stored in its long-term conceptual memory that are currently true. These instances form a belief state for the cycle in the short-term belief memory. The architecture then uses this to perform goal reasoning, where it nominates instances of its relevant top-level goals and prioritizes them according to their priority values modulated by their relevance conditions' degrees of match. Once goals are nominated, ICARUS retrieves relevant skills from its long-term skill memory and finds an executable path through the hierarchical definition of the skills and executes it in the environment. If the architecture is unable to find an executable skill path in the current state of the world, it will then invoke means-ends problem solving to find a solution to the given goal. Upon a successful problem solving, ICARUS stores the solution as a new skill in its memory for future use.

Potential Use of Generative Models in ICARUS

Given the main processes described above that occur on each of ICARUS's cognitive cycles, we now discuss ways in which generative models can be relevant and beneficial for the broader cognitive architecture. Figure 1 points out such possibilities for some architectural processes in ICARUS. While there is no doubt that we can use LLMs to greatly advance the architecture's ability to communicate through natural, linguistic means, we believe that this will be a straightforward use of those models' intended capability. For this reason, we will exclude this case from our discussion below.

Perception and Inference

There are several techniques proposed in the recent literature, especially from the computer vision community, that use generative models to process perceptual information (e.g., Wang et al., 2023b; Zang et al., 2023; Lai et al., 2023). But ICARUS's perceptual front end receives sensory input from the environment as a list of objects and their attributes, and most of these techniques would be applied beforehand, outside the cognitive architecture.

The inference process that follows, however, can benefit from such models. The traditional approach to belief inference requires pattern matching of explicit definitions of concepts against perceived information. But generative models have a generalist's commonsensical understanding of relations that they acquire from the vast amount of training data, and it may be possible to give the perceived information of objects to an LLM or even pass the input scene directly to a Visual Language Model (VLM) and get the relational instances that hold true in that state. Without explicitly defining spatial relations like one object being next to or behind another object, for example, we will be able to find what is currently true in the agent's world.

This approach can be especially effective when the object information is coming from images through some object classification system. In such cases, the information provided is typically about detected bounding boxes including the object type and the position and the size of the bounding box, and LLMs are capable of using such information. Despite the potential benefits, however, problems can arise from not having explicit definitions of relations. For instance, ICARUS often uses its concept definitions to decompose its goals into subgoals during problem solving. Hence, it will be important to have the ability to learn new relations from examples, which can come from generative models.

Goal Reasoning and Planning

Another set of processes that can benefit from the use of generative models includes goal reasoning and long-horizon planning. Agents can often have goals that are conceptually very far from their initial state, and it can be difficult to find a solution path regardless of which direction the architecture performs the search (i.e., forward or backward search). The problem is mainly due to the insufficient abstracted information an agent has, and previous work (Driess et al. 2023; Wang et al. 2023a) has shown that LLMs can be effective in guiding exploration in such cases.

For example, home assistant robots should know that a thirsty guest might want a beverage and where to go look for one, and it should bring a cloth if the guest carelessly spills the beverage on the table. To do this, the robot will need some common-sense knowledge of what kind of objects are needed in which situations and where to find such objects, and it is not feasible to encode all the information in advance. But LLMs can provide clues as to the goals and subgoals to pursue and possible sequences of actions that might work to achieve them, using the knowledge embedded in their networks.

Similarly, generative models can also help in the prioritization of goals based on the given situation. Deciding what needs to be done before others in an emergency situation can be quite tricky to a robot or an autonomous car, and the common-sense knowledge an LLM can provide will be useful while making such decisions. One example is whether to run the red lights when you are transporting patients with problems of varying seriousness in an ambulance.

But the generalist's response from LLMs might not work in all cases, and there is always the possibility of them giving unreliable answers, especially when a model suffers from the scarcity of domain-specific data online. Therefore, we argue that the use of generative models for goal reasoning and planning should be done with caution, for instance, by considering their solutions only in the absence of explicit domain knowledge.

Learning

It is ironical that LLMs are constructed through a tremendous amount of machine learning and yet they are unable to learn on the fly. One might argue that they are capable of learning during run time if we employ them in a few-shot setting (using example prompts). But the effect of this practice is temporary and the new knowledge acquired in this manner is not persistent across different sessions of LLMs.

This is where ideas from cognitive architectures, especially that of long-term memories, can help LLMs if properly integrated within a unified framework. We are already starting to see some recent work using such ideas from the LLM perspective. Wang et al. (2023a), for example, introduce a skill memory where the system can store useful procedures the agent discovers during exploration in an open-ended game. We can greatly increase the capabilities of the overall system by integrating generative models in the framework of cognitive architectures, allowing the whole suite of memory structures and cognitive processes that are common in such architectures accessible to LLMs or any other additional components that are added into them. In this regard, taking an architectural approach with LLMs seems very promising.

Conclusions

In this short paper, we noted that various LLM and generative AI applications reported in the literature go beyond simply using their linguistic capabilities and leverage language as a medium to serve other goals. Some of these include using generative models in embodied settings, and we noticed that there are no ongoing discussions on a more architectural point of view. We discussed a few ways generative models and cognitive architectures can mutually benefit from each other, in perception, inference, goal reasoning, planning, and learning. We used the ICARUS architecture as the framework for this discussion, but most of the ideas presented should carry over to other architectures equally well. We look forward to reporting on our future research along some of the directions presented.

Acknowledgments

The author would like to thank Chen Ruirui, Nguyen Thanh Son, Kenneth Kwok, Cheston Tan, Sui Xiuchao, and David Ménager for intriguing discussions that led to the ideas presented in this paper. This research was supported by Agency for Science, Technology and Research (A*STAR) under its Human-Robot Collaborative AI for Advanced Manufacturing and Engineering (Award A18A2b0046). Any opinions and conclusions expressed in this material are those of the author and may not necessarily reflect the views of the agency. No official endorsement should be inferred.

References

Anderson, J. R.; and Lebiere, C. 1998. *The Atomic Components of Thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Chen, W.; Ma, X.; Wang, X.; and Cohen, W. W. 2022. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. *ArXiv*, abs/2211.12588.

Choi, D.; and Langley, P. 2018. Evolution of the ICARUS cognitive architecture. *Cognitive Systems Research*, 48: 25–38.

Driess, D.; Xia, F.; Sajjadi, M. S. M.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q. H.; Yu, T.; Huang, W.; Chebotar, Y.; Sermanet, P.; Duckworth, D.; Levine, S.; Vanhoucke, V.; Hausman, K.; Toussaint, M.; Greff, K.; Zeng, A.; Mordatch, I.; and Florence, P. R. 2023. PaLM-E: An Embodied Multimodal Language Model. *ArXiv*, abs/2303.03378.

Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; and Iwasawa, Y. 2022. Large Language Models are Zero-Shot Reasoners. *ArXiv*, abs/2205.11916.

Lai, X.; Tian, Z.; Chen, Y.; Li, Y.; Yuan, Y.; Liu, S.; and Jia, J. 2023. LISA: Reasoning Segmentation via Large Language Model. arXiv:2308.00692.

Laird, J. E. 2012. *The Soar Cognitive Architecture*. MIT Press.

Laird, J. E.; Newell, A.; and Rosenbloom, P. S. 1987. Soar: An Architecture for General Intelligence. *Artificial Intelligence*, 33(1): 1–64.

Langley, P.; and Choi, D. 2006. A Unified Cognitive Architecture for Physical Agents. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, 1469–1474.

Talmor, A.; Tafjord, O.; Clark, P.; Goldberg, Y.; and Berant, J. 2020. Leap-Of-Thought: Teaching Pre-Trained Models to Systematically Reason Over Implicit Knowledge. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 20227–20237. Curran Associates, Inc.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L. J.; and Anandkumar, A. 2023a. Voyager: An Open-Ended Embodied Agent with Large Language Models. *ArXiv*, abs/2305.16291.

Wang, W.; Chen, Z.; Chen, X.; Wu, J.; Zhu, X.; Zeng, G.; Luo, P.; Lu, T.; Zhou, J.; Qiao, Y.; and Dai, J. 2023b. VisionLLM: Large Language Model is also an Open-Ended Decoder for Vision-Centric Tasks. arXiv:2305.11175.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; hsin Chi, E. H.; Xia, F.; Le, Q.; and Zhou, D. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *ArXiv*, abs/2201.11903.

Zang, Y.; Li, W.; Han, J.; Zhou, K.; and Loy, C. C. 2023. Contextual Object Detection with Multimodal Large Language Models. arXiv:2305.18279.