Proposal for Cognitive Architecture and Transformer Integration: Online Learning from Agent Experience

John E. Laird, Robert E. Wray, Steven Jones, James R. Kirk, Peter Lindes
Center for Integrated Cognition @ IQMRI
24 Frank Lloyd Wright Drive Suite H 1200, Box 464
Ann Arbor, Michigan 48106
{john.laird, robert.wray, steven.jones, james.kirk, peter.lindes@cic.iqmri.org}

Abstract

We explore the potential integration of Transformers, trained online in real-time using an agent's ongoing experiences, as a learning and memory component of a cognitive architecture such as Soar. We identify key challenges and potential capabilities enabled by such an integration.

Introduction

Transformers (Vaswani et al. 2017) are trained offline from large, existing corpora, with a prime example being the creation of Large Language Models (LLMs), such as GPT-4 (OpenAI 2023). LLMs have demonstrated impressive performance in retrieving contextually relevant data but do not have all the capabilities required of a cognitive agent, such as perceptual processing, decision-making, planning, and different forms of reasoning (spatial, temporal, meta). Moreover, they no do not support online lifelong persistent learning. In contrast, cognitive architectures (Kotseruba and Tsotsos 2020; Newell 1990) provide the computational infrastructure for creating embodied agents, including the ability to pre-encode knowledge and learn new persistent knowledge through various online learning mechanisms, including procedural composition, reinforcement learning, semantic learning, and episodic learning. Although diverse, those existing cognitive-architecture learning mechanisms have yet to show the ability to learn and retrieve the contextually sensitive similarity-based predictive knowledge that Transformers might afford.

This paper explores the challenges and opportunities of integrating Transformers with a cognitive architecture agent, where the Transformer is trained online with the agent's *experiences.* The Transformer becomes a memory for the agent that provides context-sensitive similarity-based predictions based on the agent's current situation. In a Transformer-based LLM, the model learns to predict the next word based on the words seen thus far and, in doing so, learns the structure of the language and how language is used. Our goal is not to predict the next word but to predict the next percept, thought, or action based not on external language produced by humans but on internal agent experiences. That is, we aim to use an LLM to learn "the structure of thought," enabling an agent to extend its reasoning to new situations and tasks while maintaining consistency with its existing knowledge and prior experiences. Recent advances in robot action-learning suggest the feasibility of capturing a subset of this knowledge (Brohan et al. 2023), in which low-level manipulation systems are trained based on observed visual and natural-language inputs to the robot.

The next section describes the desiderata of such an integration and related efforts to achieve some of those goals. Following that is an abstract description of a potential approach to realizing such an integration, a list of the technical challenges that directly relate to the desiderata, the potential capabilities such integration would engender, and a concluding discussion.

Desiderata and Related Work

Our goal is to develop long-lived, adaptive, intelligent, embodied agents that have many of the cognitive capabilities we associate with humans, including perception of complex environments, decision-making informed by large bodies of knowledge, multiple representations of knowledge, planning, meta-reasoning, communication and coordination capabilities, broad motor capabilities and multiple learning mechanisms. Below, we identify additional desiderata we consider relevant to integrating a novel learning capability based on Transformers, with commentary on how existing agent systems that use LLMs fare.

Learning is Online and Incremental

General embodied agents must adapt to novel tasks and environments through online learning, acquiring knowledge as they experience it. Given the likelihood of returning to a task, they must also maintain that knowledge throughout their existence.

Existing LLMs use offline batch learning and often adopt the goal of zero-shot learning, meaning that everything for a task is learned before the task is encountered. Although this approach may be appropriate for agents that perform limited, repeated tasks, it does not allow an agent to learn new tasks or customize its task performance to the preferences of individual human users (Kirk, Wray, and Lindes 2023).

There is a concept in the LLM literature of "in-context learning," which involves adaptation to the current situation via the prompt and previous retrievals (Brown et al.

2020). However, in-context learning does not involve persistent weight changes (it seems odd to call it learning, as it is only a temporary adaptation to the current context). Thus, although there are LLM applications where agents demonstrate sophisticated command following (Ahn et al. 2022; Yao et al. 2023), they do not learn new tasks or concepts that persist beyond an engagement with a human.

Another learning approach used with Transformers is *fine-tuning,* where the final layers of the LLM are retrained for specific tasks. However, fine-tuning is not online, and the repeated fine-tuning of the final layers risks the loss of knowledge acquired for earlier tasks.

Learning can be included by augmenting an LLM within an agent architecture as demonstrated in the Generative Agent (Park et al. 2023). The Generative Agent incorporates an additional learning mechanism and memory (specifically episodic memory) that provides persistent learning and behavioral change. The Generative Agent approach does not modify the knowledge encoded in the underlying Transformer, but the additive memories are used as persistent (retrievable) context for future queries to the LLM.

Learned Knowledge is Grounded

To be useful, the knowledge in an embodied agent must be grounded in the agent's experiences, its perception of the world, and its motor capabilities (Maher, Ventura, and Magerko 2023).

LLM knowledge is derived from language and possibly visual databases, but neither directly connects to an agent's experience. Thus, grounding is achieved through additional mechanisms (possibly other pre-trained Transformers). A challenge is extending LLMs so that their training data is more directly from agent experience and represented in the forms an agent uses for internal reasoning.

In an interesting twist, the Generative Agent "grounds" its linguistic knowledge in its environment, but only because all interactions with its environment are through language.

Learned Knowledge Correctly Transfers to New, Similar Situations

The learned knowledge should not be tied to specific experiences but generalized to apply in new, similar situations. A strength of LLMs is their ability to learn implicit, general knowledge that (usually) applies appropriately in novel but similar situations. For example, LLMs learn hierarchical and abstract syntactic structures that they can use to generate novel text. However, they also can hallucinate, generating incorrect responses, making this a significant challenge in using LLMs with autonomous agents.

Agent Responds in Realtime

The agent's computational demands are such that it can access and reason with its knowledge in real time relative to the dynamics of the environment and that its overall operation, including learning, occurs in real time.

Proposed Integration

Figure 1 shows the abstract architectural structure for the proposed integration of the Soar cognitive architecture (Laird 2012, 2022) and a Transformer.[1] Soar is representative of many cognitive architectures (Kotseruba and Tsotsos 2020) whose structure is consistent with the proposed Common Model of Cognition (Laird, Lebiere, and Rosenbloom 2017).

Figure 1 envisions a straightforward integration from an architecture perspective as the Transformer is included as a new long-term memory that operates asynchronously from the other modules with its own retrieval and learning mechanisms. Because of underlying challenges in meeting the desiderata above (see next section), in a final discussion, we explore a more complex architectural integration that requires less innovation in terms of Transformer technology.

Below are four aspects of the integration:

Transformer Input

Transformer input comes from the architecture's short-term memories. We consider only working memory for simplicity and defer considering modality-specific memories (such as from the spatial visual system). In Soar, as in many cognitive architectures, working memory is a symbolic graph structure derived from sensor data, internal reasoning, and long-term memories. It represents the agent's current situation, tasking, goals, plans, actions, communication with other agents, etc. Thus, in contrast to LLMs that focus only on language, we see language as only one component of the input to the Transformer.

One aside on the issue of language is that given that LLMs have been trained exclusively on language, one approach in other integrations of LLMs with other architectural components is to make language (words, phrases, sentences) the underlying "language of thought." That is not the approach we are taking here, and one question is whether language proves to be sufficient for all internal reasoning or whether the approach we are taking, where it is one of many possible representations, will be necessary.

Learning

Changes to working memory from perception and retrievals from long-term memories (procedural, semantic, and episodic) occur in parallel, resulting in "waves" of changes. For each wave, the Transformer will be trained to predict future changes. As noted above, these are changes to graph structures and not a linear string of tokens. As noted below, both cognitive architecture features (multiple parallel changes, changes to graph structures) raise challenges for using Transformers, requiring linearization of graphical representations (Gao et al. 2023).

[1] Transformers are singled out as a potential neural architecture; however, other neural approaches should be considered (Ororbia and Kelly 2023; Furlong and Eliasmith 2023; Smolensky et al. 2022).

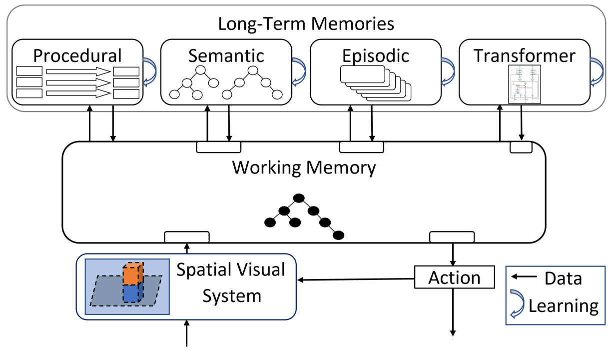

Figure 1: Proposed integration of GTN in Soar

Retrieval Prompting

Retrieval prompting for semantic and episodic memory involves the deliberate creation of a cue sent to the long-term memory. For the proposed memory, there are two possibilities that we would investigate. As with episodic and semantic memory, an agent's procedural memory can deliberately create a specific prompt (symbolic graphic structure) for retrieval to focus the memory on a particular topic. An alternative is that all working memory forms the prompt/context for retrieval. This would allow spontaneous retrieval so that the complex content of working memory (which includes prior predictions) is used to determine what is retrieved.

Transformer Output

Similar to retrievals from episodic memory, the Transformer output would be deposited in a separate buffer within working memory to distinguish retrievals from the Transformer from other sources. Procedural knowledge can access that buffer and use it for decision-making, planning, and creating other structures in working memory that influence the agent's behavior.

Challenges

Here we identify major challenges to realizing this integration approach.

Format of Transformer Input

The input for standard Transformers is a sequence of tokens, such as parts of words. Here, the input would be a symbolic, graphical structure, which, with traditional Transformers, would require linearization into a list of individual tokens. Another possibility would be to attempt to use Graph Transformer Networks (GTNs) (Yun et al. 2020; Dwivedi and Bresson 2021) as the underlying Transformer technology. Whether the assumptions underlying GTNs are compatible with this proposed use with cognitive architectures is unclear. For example, the current practice appears to train on static graph structures instead of a sequence of graph structures that can be incrementally modified over time, as in a cognitive architecture working memory.

A related issue is that cognitive architectures do not have a stream of single changes to working memory. Instead, many changes to working memory can occur in parallel. Thus, the desired behavior is to predict multiple changes to working memory from multiple changes to working memory. Once again, such changes could be linearized, but there might be unforeseen negative interactions with the operation of other cognitive architecture components, such as matching by procedural memory.

Incremental Versus Batch Training

In current practice, Transformers are batch-trained on fixed input, followed by fine-tuning. Our use of Transformers requires incremental online training. Whether this approach will converge and avoid problems such as catastrophic forgetting is an open question. Developing new online algorithms for Transformers that do not depend on batch training may be necessary. If online learning is impractical, we explore an alternative in the discussion section.

Initial Agent Knowledge

What knowledge does an agent require to exploit the capabilities of the Transformer? In the following section, we list cognitive abilities that might be realized or enhanced with a Transformer trained on experience, but how does the agent know how to use the Transformer to achieve these (and potentially other) capabilities? Is there general knowledge that can be encoded in agents to use these capabilities across whatever tasks an agent pursues? Do those capabilities emerge? Discovering and encoding that knowledge (if necessary) is an additional challenge.

Sufficiency of Experience

One lesson from LLMs is that more is better, with performance improving as the amount of data used in training and the number of parameters in the model increase. Furthermore, during training, the same data is used repeatedly.

It is an open question as to whether the singular exposure of experiences of an agent over a reasonable time will be sufficient to result in useful knowledge. Although a potentially fraught comparison, humans require extensive experience with the world, at least years, to learn complex cognitive skills. What level of experience (and timeline) can we expect from the learning we proposed?

Real-time Responsiveness

There could be challenges regarding the real-time integration of a Transformer within a cognitive architecture. The good news is that cognitive architectures like Soar are designed to support asynchronous long-term memories. Even if the Transformer is orders of magnitude slower than the inner loop of the cognitive architecture ($<$ 1 msec in Soar on modern CPUs), the overall design is robust. However, the Transformer must also keep up with the dynamics of the environment both in providing timely responses and processing input for training. A final (positive) caveat is that the total data (and thus the necessary size of the Transformer) would be many orders of magnitude smaller than what is required for LLMs.

Potential Capabilities

Transformers are prediction engines, so if such an integration succeeds, they should be able to predict the most likely future working memory states, which have many potential uses as outlined in Table 1.

Capability	Description
Anomaly detection	Detecting that a situation is no longer routine
Action modeling	Predicting the effects of an action
World modeling	Predicting world dynamics
Anticipation	Predicting details of a possible future event before it occurs
Preparation	Taking action to change an expected future

Table 1: Capabilities enabled by prediction

Existing cognitive architectures take advantage of implicit predictions (such as those encoded in memory activation and association strength) and explicit prediction through using the contents of episodic memory, but they lack three potential differentiators that Transformers might provide:

1. Broader, more complete contextual retrieval that a Transformer could provide.
2. Spontaneous retrieval without explicit prompting (ACT-R provides comparable functionality in some cases).
3. Implicit, similarity-based generalization and synthesis of past experiences that predict the future.

These capabilities do not negate the value or necessity of episodic memory as it provides *instances* of experience, whereas a Transformer would provide generalization and synthesis over those experiences. Instances are critical for answering questions about what actually happened in the past. In contrast, the synthesis provided by Transformers could support predicting future states, even from states that were never experienced. On the surface, these may seem like distinctions without important differences; however, episodic memory allows an agent to remember exactly what commitments it has made in the past to another agent, whereas synthesis enables an agent to predict, in general, the type of commitments it (or possibly others) would make in the current situation.

Beyond the prediction capabilities envisioned in the table, it is difficult to anticipate all the capabilities an integrated Transformer-based memory and learning system could engender. One of the lessons we take away from LLMs is how the field has found uses for them that, although grounded in the LLM's prediction capability, appear qualitatively different than a prediction. There is the straightforward use of them as a knowledge source (Kirk, Wray, and Laird 2023), similar enough to long-term semantic memory that it might replace it (in contrast to episodic memory, as noted above). Similarly, a Transformer could provide entailments of the current situation (called elaborations in Soar) that enrich situational awareness for the agent. However, some capabilities involve creating complex, extended structures, such as writing a summary or critique of an article, providing task plans, and others. An implication appears to be that improved prediction in language supports a wide variety of cognitive functions. One of our hopes is that prediction, grounded in an agent's experience in addition to the language it uses and hears, will offer similar functional advantages and opportunities for the agent.

Discussion

Given the uncertainty of the ability to create graphical Transformers that can robustly and incrementally learn from graphical data, it may be worthwhile to explore alternative architectures to achieve learning from experience with a Transformer. To simplify the description, imagine a robot that is active for 8 hours a day. During that time, it relies on previous training of its Transformer while depending on other architectural learning and memory mechanisms and context learning for short-term adaptation. During inactive time (sleep), the agent could replay its experiences stored in episodic memory to fine-tune the Transformer. At an abstract level, there appear to be parallels between these processes and consolidation in humans during sleep. When the robot has a longer period of inactivity, it could retrain using episodic memory (not a process found in humans). Answering the question as to the ratio of active experience to the time required for fine-tuning and retraining downtime would influence the practicality of this approach in real-world scenarios.

One intriguing possibility is that this approach could be completely under the agent's control. It would require some deliberate control of initiating fine-tuning and training of the Transformer by feeding it reconstructed experiences from episodic memory. This approach might be a preferred starting point as it would provide a baseline and research experience with GTNs, fine-tuning, and retraining before confronting real-time online learning of a Transformer.

Acknowledgments

This work was supported by the Office of Naval Research, contract N00014-21-1-2369. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Department of Defense or Office of Naval Research. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

Ahn, M.; Brohan, A.; Brown, N.; Chebotar, Y.; Cortes, O.; David, B.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; et al. 2022. Do As I Can, Not As I Say: Grounding Language in Robotic Affordances. In *6th Annual Conference on Robot Learning*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Choromanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; et al. 2023. RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control. *arXiv preprint arXiv:2307.15818*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language models are few-shot learners. *arXiv preprint*, arXiv:2005.14165.

Dwivedi, V. P.; and Bresson, X. 2021. A Generalization of Transformer Networks to Graphs. ArXiv:2012.09699 [cs].

Furlong, P. M.; and Eliasmith, C. 2023. Vector Symbolic Algebras Underpin Generative Cognitive Models. In *Proceedings of the 2023 AAAI Fall Symposium on Integrating Cognitive Architectures and Generative Models*. AAAI Press.

Gao, Z.; Zhang, S.; McLoughlin, I.; and Yan, Z. 2023. Paraformer: Fast and Accurate Parallel Transformer for Non-autoregressive End-to-End Speech Recognition. arXiv:2206.08317.

Kirk, J. R.; Wray, R. E.; and Laird, J. E. 2023. Exploiting Language Models as a Source of Knowledge for Cognitive Agents. In *Proceedings of the 2023 AAAI Fall Symposium on Integrating Cognitive Architectures and Generative Models*. AAAI Press.

Kirk, J. R.; Wray, R. E.; and Lindes, P. 2023. Improving Knowledge Extraction from LLMs for Robotic Task Learning through Agent Analysis. ArXiv:2306.06770 [cs].

Kotseruba, I.; and Tsotsos, J. K. 2020. 40 years of cognitive architectures: core cognitive abilities and practical applications. *Artificial Intelligence Review*, 53(1): 17–94.

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

Laird, J. E. 2022. Introduction to Soar. ArXiv:2205.03854 [cs].

Laird, J. E.; Lebiere, C.; and Rosenbloom, P. S. 2017. A Standard Model of the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics. *AI Magazine*, 38(4): 13–26. Number: 4.

Maher, M. L.; Ventura, D.; and Magerko, B. 2023. The Grounding Problem: An Approach to the Integration of Cognitive and Generative Models. In *Proceedings of the 2023 AAAI Fall Symposium on Integrating Cognitive Architectures and Generative Models*. AAAI Press.

Newell, A. 1990. *Unified Theories of Cognition*. Cambridge, Massachusetts: Harvard University Press.

OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774.

Ororbia, A.; and Kelly, M. A. 2023. A Neuro-Mimetic Realization of the Common Model of Cognition via Hebbian Learning and Free Energy Minimization. In *Proceedings of the 2023 AAAI Fall Symposium on Integrating Cognitive Architectures and Generative Models*. AAAI Press.

Park, J. S.; O'Brien, J. C.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative Agents: Interactive Simulacra of Human Behavior. ArXiv:2304.03442 [cs].

Smolensky, P.; McCoy, R.; Fernandez, R.; Goldrick, M.; and Gao, J. 2022. Neurocompositional Computing: From the Central Paradox of Cognition to a New Generation of AI Systems. *AI Magazine*, 43(3): 308–322.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. In *31st Conference on Neural Information Processing Systems*, NIPS, 6000–6010. Long Beach, CA: Curran Associates Inc. ISBN 978-1-5108-6096-4.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. arXiv:2210.03629.

Yun, S.; Jeong, M.; Kim, R.; Kang, J.; and Kim, H. J. 2020. Graph Transformer Networks. ArXiv:1911.06455 [cs, stat].