Research Statement: Knowledge Search in Cognitive Architectures

Justin Li

As the applications of artificial intelligence extend into knowledge-rich domains, agents need access to increasingly diverse sources of knowledge – general knowledge from Wikipedia, linguistic knowledge from WordNet, even knowledge of the agent's own past experiences in an episodic memory store. While there has been a lot of work optimizing knowledge-base queries for performance [1, 2], the questions of how agents should use these knowledge bases and when they should do so remain unexplored. These agents may not know what knowledge can aid in their problem solving, nor whether such knowledge exists; in contrast, IBM's Jeopardy-playing Watson only searches its knowledge base when asked a question, and that question is guaranteed to have an answer. The broad theme of my research is therefore solving the problem of *knowledge search*: how an agent finds relevant knowledge at the right time for use in decision making.

Consider an agent with access to a knowledge base of research papers, including a paper on reinforcement learning for accessing knowledge bases. Amidst a stack of other papers, the agent finds a paper on learning memory search strategies. While this should prompt the agent to query its knowledge base for connections, this raises questions about the processes of the agent. How does the agent know that, of all topics, its knowledge base has additional information about learning memory strategies? It would be time consuming to search the knowledge base for every paper, particularly since "learning" and "memory" are common keywords and would return superfluous results. Furthermore, "learning" does not directly imply "reinforcement learning", nor does "memory" imply "knowledge base"; additional inference is necessary for the agent to reformulate its query. In short, the agent faces the problem of deciding whether and how to perform a knowledge search.

My thesis specifically addresses when the agent should search its knowledge base, in the restricted setting of *goal re-activation*: how does an agent detect that a previously inactive goal, stored in a knowledge base, is now suddenly relevant? I developed new knowledge-access mechanisms and corresponding strategies for this problem, and also determined the policy for selecting between strategies. With this synthesis of computer and cognitive sciences, I plan on extending these strategies to the larger knowledge search problem, to build agents that can find the necessary knowledge at the right time across diverse domains and knowledge bases. This work is one step towards the overall goal of building robust, generally intelligent agents that can perform a variety of knowledge-intensive tasks.

Thesis Work

If the agent has a goal (among many) of giving a message to a colleague, and the agent does not know when it will see the colleague, how does the agent detect that the goal is now relevant when the colleague is in sight?

My thesis answers this question in the context of research in cognitive architectures, which aims to define and understand the computational processes and mechanisms that enable human-level intelligence, including standardized mechanisms for retrieving knowledge from long-term knowledge stores. For computational complexity reasons, many cognitive architectures partition knowledge into long-term and short-term knowledge stores (*long-term memory* and *working memory*, respectively), only the latter of which can be used in decision making. To transfer knowledge from long-term memory to working memory, the agent must describe the features of the desired piece of knowledge in a *deliberate cued retrieval*, after which the piece of knowledge that best matches that description is placed in working memory and can be used for reasoning. Since the majority of goals do not immediately contribute to reasoning, they are stored in long-term memory; the problem of goal re-activation asks when goals should be retrieved into working memory.

Goal Re-activation: Strategies for Breaking Circular Knowledge Dependencies

Agents developed with a cognitive architecture have a circular knowledge dependency problem when they try to re-activate goals. Consider the message-giving agent when it meets its colleague. Since the agent does not have direct

access to long-term memory, the agent cannot directly detect that the goal is relevant – that is, the agent would not immediately know that the colleague is the achievement condition of a goal, unless it searches long-term memory for knowledge about that colleague. At the same time, long-term memory retrievals must be deliberately initiated, and without knowing that a goal is relevant, the agent has no reason to do so. This is creates a circular dependency: goals that are stored in long-term memory must be retrieved before their relevance can be assessed, but it is the knowledge of a relevant goal that triggers the agent to retrieve the goal in the first place. This is exacerbated by the cost of searching its knowledge base, which grows with both the number of goals and with the number of changes in perception; this makes the naive strategy of searching at every time step costly [5].

Drawing inspiration from the prospective memory literature on how humans complete delayed goals, I initially defined two strategies for breaking the dependencies. The first strategy stores goals as if-then rules that automatically match and modify working memory; this spares the agent from needing to retrieve goals. A human analogy might be to repeatedly give messages to the colleague, which eventually turns the behavior into a habit or a reflex. This strategy is unsuitable for one-off tasks like giving a message; in particular, the agent would keep giving its colleague the message every subsequent time the colleague is nearby. The second strategy periodically retrieves all goals from long-term memory to check for their relevance; this uncouples the retrieval of the goal from the determination of its relevance, breaking the knowledge dependency cycle. In people, an equivalent strategy might be reminding yourself of all your goals at regular intervals. The retrieval interval must be tuned such that it is neither too frequent (thus wasting resources checking every goal) nor too infrequent (thus missing opportunities to achieve goals).

To evaluate of these strategies, I created an environment that represents goal re-activation tasks in the abstract, where goals and their conditions are probabilistically generated. The agents are evaluated not only on the proportion of goals they complete, but also how efficiently they complete them. In this domain, I showed that these two strategies trade off in their performance. While both strategies allow the agent to complete goals, they incur different costs: the habit-based strategy fails to delete completed goals, while periodic retrievals require resources proportional to the length of the delay. This suggests that effectively completing delayed goals requires new knowledge-access mechanisms [4, 6].

Knowledge-Access Mechanisms: Metamemory Judgment and Spontaneous Retrieval

A different link in the dependency cycle could be broken by automatically retrieving goals into working memory when they might be relevant, bypassing the need for agent deliberation. The prospective memory literature suggests that this occurs with humans: people have "feelings of knowing" that they have to do something, and goals may spontaneously "pop" into mind. Since these automatic mechanisms have never been explored in cognitive architectures, in my thesis I designed and implemented two such mechanisms, and showed that they not only aid goal re-activation, but also provide functional benefits in other domains.

The first mechanism is metamemory judgment, which automatically provides the agent with metadata about the contents of long-term memory, such as whether the current percepts have been seen before. This mechanism exposes information in the indices that already exist for efficient memory retrieval, and is therefore computationally efficient. It allows the agent to use the metadata as a signal to retrieve its goals, instead of having to determine the relevance of a goal before retrieving it, thus breaking the dependency cycle; this mirrors the "feeling of knowing" that might occur in people when the target colleague appears. This novel mechanism is useful not only for goal re-activation, but also for reducing the number of memory retrievals in other domains, by acting as an inexpensive predictor of whether knowledge exists. In a word sense disambiguation task, I showed that this mechanism allows the agent to reduce the number of searches by between 18.8% and 68.1%, with minimal reactivity cost while retaining the same level of performance [3].

The second mechanism I designed is spontaneous retrieval, which provides the agent with a complete piece of knowledge that is relevant to the agent's current situation, heuristically determined using the agent's memory-search history. In goal re-activation, goals can be spontaneously retrieved if the achievement conditions are met, similar to how goals "pop" in people's minds. Spontaneous retrieval thus allows the use of knowledge in long-term memory

even if the agent does not retrieve that information or does not know that it should be retrieved. This is demonstrated in the Missing Link word puzzle, where the likelihood that a solution exists is parameterized; since the agent does not know whether a particular puzzle has a solution, it must exhaustively search memory before giving up. With spontaneous retrieval, the agent merely has to verify whether a retrieved item is the correct answer before using it as its solution. This agent takes 17.9% to 86.7% less time to complete a series of puzzles, even when 50% of the puzzles are unsolvable [7].

Although the implementation of goal re-activation strategies that use these mechanisms is still in progress, initial analysis suggests they can avoid the costs incurred by strategies that only use existing memory mechanisms.

Future Work

My thesis focused on determining whether relevant knowledge exists in long-term memory; translated to the artificial research assistant example, it is equivalent to recognizing that a previous paper may be relevant. However, my thesis assumes that relevant knowledge can be directly retrieved, when in fact determining relevance may be non-trivial, such as the conversion from the general idea of "learning" into a specific "reinforcement learning" algorithm. In addition to strategies for determining when to search memory, agents would also need strategies for *how* to search memory, as well as the ability to adopt different strategies depending on environmental demands. For my future work, I plan on expanding the repertoire of memory strategies to allow robust use of knowledge from all available knowledge sources.

Knowing How to Search: Knowledge Search Strategies

Knowledge search strategies apply problem solving to the goal of retrieving a useful piece of knowledge. For example, generalizing how the agent may to go from considering "learning" to searching for "reinforcement learning", one strategy may be to search memory for all subclasses of a relevant property. The general problem is that no representation of knowledge could contain all keyword indices for all future situations, especially when knowledge is incomplete and dynamically changing. A major challenge in this research is that the agent may not be able to describe the desired knowledge; the agent may only be able to verify that the knowledge is useful after it has been found. A knowledge search strategy must therefore rely on more abstract properties of knowledge, such as ontological information, to transform the problem state into one on which the agent could make progress. Drawing on logical inference and knowledge representation research, I believe advances in this area will result in strategies that can be transferred across domains with different knowledge structures, allowing all agents to more efficiently find decision-making knowledge.

Knowing When to Search: The Utility of Information and Metamemory Control

As my work on goal re-activation showed, it is not enough for the agent to know how to find information, but must also know when to do so and when to give up: the best strategy may be to not search at all. The memory mechanisms that I designed provide the agent with more information about the state of its memory, allowing it to avoid the extreme cases of searching when there's nothing to be found, and not searching when something can be found. The use of this information to direct knowledge search remains unexplored. For example, to use the "learning" example above, which subclass of learning should be explored first, and when should the next subclass be explored instead? Although a full search of memory may be necessary to find the best answer, the cost of doing so may not be worth the benefits of getting it precisely right. This utility tradeoff exists not just in searching memory, but also in knowledge search of external information sources. Models of these information seeking behaviors – like that of information foraging, the idea that information exists in clusters with parallels to food foraging behavior – have only been used to model humans, but has not been applied to artificial agents. With knowledge of both when to search for knowledge and how to do so, the agent has the capability to bring to bear any resources it can find in achieving its tasks.

References

[1] Nate Derbinsky, Justin Li, and John E. Laird. Algorithms for scaling in a general episodic memory (extended abstract). In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1387–1388, 2012.

[2] Nate Derbinsky, Justin Li, and John E. Laird. A multi-domain evaluation of scaling in a general episodic memory. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, pages 193–199, 2012.

[3] Justin Li, Nate Derbinsky, and John E. Laird. Functional interactions between memory and recognition judgments. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, pages 228–234, 2012.

[4] Justin Li and John E. Laird. Preliminary evaluation of long-term memories for fulfilling delayed intentions. In *AAAI 2011 Fall Symposium on Advances in Cognitive Systems*, pages 170–177. AAAI Press, 2011.

[5] Justin Li and John E. Laird. The computational problem of prospective memory retrieval. In *Proceedings of the 12th International Conference on Cognitive Modeling (ICCM)*, pages 155–160, 2013.

[6] Justin Li and John E. Laird. Preemptive strategies for overcoming the forgetting of goals. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI)*, pages 1234–1240, 2013.

[7] Justin Li and John E. Laird. Spontaneous retrieval from long-term memory for a cognitive architecture. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*, 2015. (In publication).