Levels of AI Memory—And Case-Based Ways for LLMs to Ascend Them

Michael W. Floyd[1], David Leake[2], David H. Ménager[3], Ian Watson[4] and Kaitlynne Wilkerson[2]

[1]*Knexus Research LLC, National Harbor, MD, USA*
[2]*Luddy School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA*
[3]*AI and Autonomy Group, Parallax Advanced Research, Beavercreek, OH, USA*
[4]*School of Computer Science, University of Auckland, Auckland, New Zealand*

Abstract

The topic of memory has long been a motivation and central focus in the area of case-based reasoning (CBR), influencing how cases are obtained, stored, retrieved, updated, and even forgotten. But memory is not a topic exclusive to CBR-based AI systems and is, in fact, necessary for AI systems to operate in many use cases. In this paper, we more generally examine the topic of memory in AI systems, presenting our *Six Levels of AI Memory* framework and describing the types of reasoning each level supports. We use these levels of AI memory to frame discussion of the existing and future memory capabilities of CBR and large-language models (LLMs), providing insight into how CBR's history of memory-focused reasoning can support LLMs in enhancing their memory.

Keywords

Case Based Reasoning, ChatGPT, Cognitive Science, Large Language Models, Levels of Memory, Episodic Memory, Retrieval Augmented Generation

1. Introduction

Large Language Models (LLMs) have grasped public attention since the release of ChatGPT in late 2022 and have had transformative impact on AI. Since then, individuals, researchers, and industry have widely applied LLMs for a growing range of tasks, such as planning [1], question answering [2], and reasoning [2, 3]. To be able to operate effectively in these contexts, a range of memory capabilities will be required. While current LLMs have been augmented with varying memory capabilities, the need for robust memory remains, and its development has been highlighted as a key challenge [4].

Bach et. al [5] present a broad range of opportunities and challenges for integrations of CBR and LLMs, including the potential for applying CBR-inspired methods to improving LLM memory. Of necessity, that paper presented key points at a high level. This paper elaborates on the themes presented there for integration of episodic memories inspired both by the cognitive

International Conference on Case Based Reasoning '25: Case Based Reasoning and Large Language Models Synergies Workshop, June 30, 2025, Biarritz, France
*Corresponding author.
The authors contributed equally.
0000-0002-8666-3416 (D. Leake)

models underlying CBR and by the memory strategies and algorithms developed in CBR research. Henceforth, we will refer to these models as "CBR-inspired memory."

We begin the paper by introducing a hierarchy for organizing AI memory capabilities, which we call the Six Levels of AI Memory. This framework provides a breakdown of the ways in which AI systems may remember, store, use, and forget information, and is designed to facilitate discussion on memory capabilities. We then provide a brief introduction to cognitive science research giving rise to theories of human memory that shaped CBR memory models and elaborate, refine, and extend the opportunities from [5] using the Levels of AI Memory framework as well as the techniques developed by the CBR community for handling memory. Finally, we examine the challenges for implementation of CBR inspired memory and LLMs to achieve specific levels in the Six Levels of AI Memory framework.

2. Levels of Memory

To our knowledge, no categorization of the spectrum of memory levels on which AI systems operate has been defined. We propose the Six Levels of AI Memory framework, analogous to the levels of autonomous driving [6]:

Level 0: No Memory: Some AI systems may reason exclusively on the current set of inputs and make no attempt to leverage prior information. When the current input x_i is provided to the AI system, the output y_i is only a function of the input: $y_i = f(x_i)$. As an example, a static classifier such as a neural network with an unchanging structure or parameters has no memory; the outputs are only dependent on the current inputs. Regardless of the user or context, these systems will always produce the same output given the same input (assuming no non-determinism in their process). These types of AI systems can be thought of as being *reactive* because they do not maintain any state information for subsequent reasoning or attempt to learn. This level of memory requires all relevant information to be fully specified in the current input—otherwise the AI system will not have sufficient information to reason successfully.

Level 1: Short-Term Working Memory: This level is the first at which an AI system has some level of memory that is used during its reasoning. Short-term working memory involves using both the current input and a fixed number of prior inputs when producing output, providing a limited amount of contextual information that can be used by the AI system. For a memory of length n, the output y_i is a function of the current input x_i and the n prior inputs $x_{i-1}, \ldots, x_{x-n}$: $y_i = f(x_i, x_{i-1}, \ldots, x_{i-n})$. This level of memory overcomes the limitation of having no memory in that not all relevant information needs to be in the current input; relevant information can also be in the prior inputs. However, because short-term working memory uses a fixed-length window, there is no guarantee that relevant information will be within that window. Similarly, there are no guarantees that the short-term memory window only includes inputs relevant to the current context. For example, an input window of length 5 may have inputs from multiple users or multiple applications, and only the most recent user or application may be relevant.

Level 2: Session Memory: This level of memory is similar to short-term working memory in that it allows an AI system to reason using information from prior inputs but differs in

how many prior inputs are provided. Whereas short-term working memory always uses a fixed-length window of past inputs, session memory uses a variable-length window that allows it to better align with the current context. When a session begins at time j, the input at that time x_j serves as the start of the dynamic window and extends to the current input x_i ($i \geq j$). Thus, at the start of a session (when $i = j$), no prior inputs will be included but the length of the window will grow as more inputs are received and the distance between i and j increases: $y_i = f(x_i, x_{i-1}, \ldots, x_j)$. For example, when a new user begins using an AI system or an AI system is used for a new task, a new session could be created to account for the contextual change. This type of memory allows AI systems to better align the inputs they reason with to their current context by only considering inputs that occurred during the session. Comparing it to short-term working memory, it avoids the possibility of inputs from multiple contexts being part of the same window, which could potentially cause issues during reasoning (e.g., the current user may not want the system to reason using inputs from the previous user). However, session memory relies on the relevant session inputs occurring in sequence during an uninterrupted window. If *User A* uses an AI system and then takes a break while *User B* uses it, a new session would be created when *User A* returns to the system, forgetting all of their previous inputs from their initial session.

Level 3: Long-Term Memory: The previously described levels of memory make information available over pre-defined time periods (e.g., fixed windows or sessions) but do not provide longer-term access to that information later. After the time window or session end, the information is discarded and no longer available for future reasoning. One option would be to make the memory windows or sessions arbitrarily long, such that outputs are a function of all previous inputs: $y_i = f(x_i, \ldots, x_0)$. However, over the lifetime of an AI system the number of inputs is likely to become very large, making such an approach computationally infeasible. Additionally, it does not overcome the challenge that not all past inputs are relevant for the current context. Instead, long-term memory provides the ability to store relevant information in a persistent form such that it can be retrieved and used later. This idea is central to case-based reasoning, where the case base serves as a long-term memory that can be added to and retrieved from as needed. Similarly, retrieval-augmented generation (RAG) (e.g., [7]) leverages elements of long-term memory in storing information in outside data structures and retrieving that information to supplement the current inputs (although LLMs do not natively self-store additional information into their long-term memory during reasoning). Thus, reasoning is a function of both the current input x_i (and possibly previous inputs if short-term working memory or session memory is used) along with information from the long-term memory $\mathcal{L}$ (e.g., a case base or RAG documents): $y_i = f(x_i, \mathcal{L})$. Additionally, long-term memory could be further subdivided based on the user (e.g., only retrieving information from a specific user) or task (e.g., only retrieve information related to someone performing a specific task, sub-task, or goal). Long-term memory begins to provide AI systems with the lifelong ability to learn and reason that makes them suitable for applications that undergo regular context shifts or require longer-term recall of relevant reasoning information.

Level 4: Human-Like Memory: While long-term memory provides the ability to store and retrieve information over the lifetime of an AI system, the memory structures used are often

coarse and do not mimic the nuanced and dynamic memory exhibited by humans. As an example, while long-term memory may support storing one or more of a user's previous sessions, it may lack relevant contextual information such as why the session was performed (e.g., goals), what the results of the session were (e.g., success, failure, partial success), key takeaways from the session (e.g., difficulty performing certain tasks), and other relevant information. Human-like memory, on the other hand, supports such richer representations, making it easier to extract relevant insights from memory or supporting transfer of memories to new contexts (e.g., using the memory of booking a visit with the doctor to book a visit with a science tutor). Similarly, attaining the human-like memory level includes improving the way in which memories are represented and structured. This can include aspects such as organizing memories in a hierarchical representation, clustering related memories together, building graph-like structures to represent memories and how they relate to each other, and decomposing memories into sub-units. Aspects of human-like memory have often influenced case-based reasoning research to improve the storage and retrieval of cases (as discussed in Section 3) but this level of memory is still lacking in LLMs.

Level 5: Super-Human Memory: Human-like memory helps bring AI systems in line with the memory capabilities of humans performing the same task but may also suffer from some of the same shortcomings. For example, human memory is not infinite so some information is routinely forgotten, similarly to how an AI system may need to forget information due to storage limitations. In some situations, forgetting is beneficial for both humans and AI systems if the memories that forgotten are no longer relevant or needed (e.g., forgetting the password to a computer you no longer own), or replacing many similar specific memories with a generalization may be useful. However, in other situations valuable information may be inadvertently forgotten (e.g., forgetting the password to a computer you have not used in several years). Similarly, the ways in which memories are stored or organized may result in an overgeneralization that keeps high-level concepts but loses low-level specifics (e.g., remembering how to drive a car but not how to access the engine of a specific model of car). Going past the capabilities of humans, super-human memory would support the ability to accurately store and retrieve an unlimited set of memories over the lifetime of an AI system without making it computationally infeasible. Super-human memory may also support memory sharing, where multiple AI systems can share information in a common memory without having to explicitly communicate with each other. Such memory would provide an interconnected network of AI systems with immediate access to the shared experiences of all systems within the network. This would be similar to immediately knowing how to solve a technical issue with a computer if someone in your network had previously encountered and solved that problem (i.e., without having to explicitly ask them or search for a solution). Federated memories might provide the capability to access experiences gathered from different perspectives. While those are three examples of super-human memory capabilities, the scope of super-human memory is broad and includes anything that provides an AI system with memory that improves upon human capabilities. Although many aspects of super-human memory may be infeasible with current AI systems, they provide a general target for how such systems can surpass human memory and provide functionality that exceeds the capabilities of a human, or even a group of humans.

LLMs: Current-generation LLMs operate primarily at Levels 1 and 2, with elements of Level 3 (e.g., using RAG) [7]. For example, large commercial models like Google Gemini or ChatGPT support short-term or session-level memory by maintaining a context of previous queries and responses to the user. The primary differentiator between LLMs that operate at Level 1 or Level 2 memory is that LLMs at Level 1 only support using past interactions as contextual information whereas LLMs at Level 2 allows starting (and stopping) new sessions. By explicitly starting new sessions, these LLMs are able to reset their memory and start back at a "blank slate". However, even commercial LLMs that support very large context windows (e.g., Google Gemini supporting millions of tokens for input), there is still a limitation on the short-term and session-level memory available. Even for these advanced commercial LLMs, the stored interactions are only temporary and are never explicitly moved into a longer-term memory. Levels 3 and 4 would allow LLMS to support storing/recalling past experiences, improving dialogues, and learning (outside of just retraining/fine-tuning). Progressing through memory levels requires implementing long-term storage, complex associations, dynamic memory models, and enhanced recall/sharing. Higher levels unlock benefits, such as personalization, coherent responses, and advanced reasoning, leading to AI systems accurately modeling and surpassing human memory (e.g., Level 5).

3. Dynamic Memory and CBR

One trigger for the study of case-based reasoning was study of human *remindings* and their role in understanding and learning: Schank's Dynamic Memory Theory [8]. In Dynamic Memory theory, the same structures that guide understanding—Memory Organization Packages, or MOPs—organize memories. MOPs are hierarchical and memory is reconstructive, with memories of an episode collected by assembling the components indexed under MOPs for their constituent parts. Remembering a doctor's visit, for example, could involve assembling components for MOPs for check-in, waiting in the waiting room, the doctor's examination, and payment. The waiting room MOP can be shared by other contexts, such as a lawyer visit. Consequently, learning in one context is naturally available in the other—in this way, the model provides cross-contextual learning. In addition, Dynamic Memory theory hypothesized the existence of abstract thematic organization points (TOPs) organizing episodes by abstract features such as configurations of goals (e.g., for *Romeo and Juliet*, Mutual Goal; Outside Opposition). Indexing by explanations of failures and by TOPs accounted for a wide range of cross-contextual remindings. As we discussed in the previous section, many of these features of dynamic memory are necessary for AI systems, particularly LLMs, to advance to human-like and super-human memory.

Research on Dynamic Memory theory gave rise to a series of computer models [9] including some of the first CBR systems. These included models of memory search [10], MOP generation during understanding [11], failure-driven reminding [12], and case-based reasoning for tasks such as subjective assessment, planning, and parsing (for an overview see [13]). Much of this work focused on memory organization, and specifically on developing indexing vocabularies [14, 15, 16]. As described in Section 4, such methods can form a potential basis for external case memories to support LLMs by providing episodic knowledge. However, they also involve

1. Providing input knowledge for LLMs
    - Query anticipation/completion
    - Context-focused RAG
    - Long-term personalization
2. Supporting capture and reuse of prior LLM processing
    - Supporting response consistency
    - Supporting solution accuracy
    - Error anticipation/avoidance
    - Speedup through reuse and adaptation
3. Supporting persistent dynamic memory for continual updating
4. Supporting cross-context knowledge access (applicable to all previous items)
5. Supporting exchange of memories between LLMs

List 1
Memory opportunities (cf. Bach et al. [5])

challenges, which we delineate in Section 5.

4. Opportunities from CBR-Inspired Memory Systems for LLMs

Bach et al. observed that "generation of useful abstractions and larger-scale knowledge structures can provide the basis for new functionality for LLM users." That work proposes various functionalities that CBR-based memories could provide for LLMs, which we revise and augment to form the items of List 1. This list presents 5 types of opportunities, retrieving information to provide as inputs to support interactions with LLMs, improve LLM accuracy, and enable long-term personalization; supporting capture and reuse of LLM processing, supporting continual updating, supporting cross-contextual knowledge access, and supporting exchange of memories across LLMs. In the remainder of this section, we present each in more detail.

4.1. Providing input knowledge for LLMs

A memory that stores traces of user interactions can be used to anticipate user prompts based on prior experience, to facilitate interactions through query anticipation and prompt proposal/-completion. As an example, typical users of a travel system may first ask questions about flight options to a destination, followed by questions about hotels, and then finally questions about restaurants and activities. Being able to identify that a user is following a common usage pattern stored in the system's memory and proactively providing them information would improve the user's experience. Using information from past sessions (i.e., Level 2 memory) or long-term knowledge allows AI systems to leverage their memory by understanding a user's particular context in a meaningful way.

Retrieval-augmented generation (RAG) is an effective method to improve LLM performance, by prompting the LLM with additional domain knowledge [7, 17, 18]. Often, the provided

knowledge is general factual knowledge about the domain. However, providing well-chosen cases instead of general knowledge can be advantageous [19, 20]. In addition to simply retrieving query-relevant cases, the right memory model could retrieve context-relevant cases, organized by active understanding structures, as in Dynamic Memory theory, for *Context-focused RAG*.

If retrieval is shaped by a dynamic memory able to learn new knowledge structures from processing particular types of queries, and to understand the underlying goals of those queries and how they relate to particular users, the memory will also be able to learn new contexts and to retrieve accordingly. If memory reflects the agent seeking the information in its organization scheme, retrievals may be tailored to user needs, for *long-term personalization*.

4.2. Supporting capture and reuse of prior LLM processing

The availability of episodic memory can support trust, accuracy, and efficiency. One factor affecting trust in AI systems is consistency (i.e., whether they respond in predictable ways to new situations). However, the stochastic nature of LLM responses results in varying answers for a single prompt. A CBR system can be wrapped around the LLM process as an intelligent component [21] to capture prompts and solutions for reuse, bypassing LLM reasoning from scratch for similar future problems. Such an approach can *support response consistency*, replacing stochastic solution generation with a deterministic process. This is comparable to a group of students each asking their teacher for details about an upcoming exam. Ideally, the teacher should provide similar responses to all students and ensure that they are not providing any erroneous or contradictory information to some students.

When generating solutions from cases, a system can benefit from past feedback, such as noting cases that have been vetted as correct and trustworthy. Likewise, the level of similarity required for invoking CBR, and the allowed level of case adaptation, can constrain the reuse of memories to *support solution accuracy* from the CBR component, with users notified of whether the solution is based on existing solutions or generated from scratch. Returning to the student-teacher example, having cases related to responses which were found to be particularly beneficial for student comprehension would allow those cases to be reused with students that have similar educational needs.

An early observation of CBR is that cases are valuable for learning from both successes and failures (e.g., [22]), and that this enables *anticipation and avoidance of failures* [23]. Retrieved past cases that report LLM failures can identify and warn of those failures in the future, and, if a correct solution was generated in response to the failure in the past, that solution is available to replace the erroneous one. This is especially important given the number of LLMs that are available, each with their own strength and, more importantly, weaknesses. Since the weaknesses of a particular LLM may not be fully understood at its release, a memory of when the LLM failed and how to correct failures (e.g., "Provide an example to the LLM along with your question") can improve long-term performance. Similarly, a CBR system could remember which LLMs are better suited for particular queries and intelligently route queries to an appropriate LLM.

An early motivation for CBR was speedup learning, which aims to avoid the wasted effort of re-generating solutions from scratch when a prior solution could be adapted more efficiently. Solution generation by LLMs is expensive and its cost is a growing concern (e.g., [24]). Replacing

LLM inference with CBR, where possible, could potentially improve both speed and energy efficiency of inference, especially for tasks in circumscribed domains for which small case bases suffice; potentially in systems with multiple small cases that can bring expertise to particular task domains [25]. As an example, a common question such as "What is the capital of France?" may not require full RAG-based inference using a large commercial model.

4.3. Supporting persistent dynamic memory for continual updating

Because memories in a Dynamic Memory are reconstructed, by using the knowledge structures used for understanding, changes in those knowledge structures enable the memories to naturally reflect intervening learning. Likewise, memories are reorganized and re-indexed as the memory is used. This provides both a persistent memory and the capability for flexible updating. Over time, stored cases may reflect richer information, and prompts and stored solutions may be adjusted dynamically. CBR research has developed an extensive set of strategies for maintaining case bases, such as deleting and updating cases [26]; some of these are discussed in Section 5.

4.4. Supporting cross-context knowledge access

The sharing of knowledge structures across contexts, combined with the indexing of episodes under those knowledge structures, enables cross-contextual remindings. Likewise, abstract indexing structures such as TOPs can characterize episodes based on thematic similarities and retrieve useful cases despite surface differences. For example, CBR-inspired memory has been applied in educational contexts, using rich abstract indexing structures, to support tasks such as the retrieval of stories based on deep thematic similarities, even if they might have divergent surface features [16]. The capability to access knowledge across context increases the potential pool of useful knowledge to bring to bear for reasoning and learning.

4.5. Supporting exchange of memories between LLMs

Case-based reasoning research includes studies of distributed CBR, including how federations of CBR systems can exchange memories [27]. Methods have been developed to federate CBR systems to achieve the benefits of case sharing while minimizing the amount of shared data [28], for determining when to dispatch retrieval queries to other case bases and which adaptations are needed based on inter-case-base differences [25], and for reconciling heterogeneous case representations in multi-case-base systems [29].

5. Challenges in Implementing CBR-Inspired Memory

Having presented our taxonomy of the levels of memory and discussed opportunities for combining CBR-inspired memories with LLMs, we now consider the challenges of constructing such systems. For each challenge, we make contact with the levels of memory taxonomy and provide suggestions for how to overcome the challenges at each level. We begin by discussing issues with memory maintenance and scalability. Then, we move to discussing challenges with retrieval and adaptation.

Any design for memory-enabled LLM systems should ensure that the memory requirements do not exceed the physical limits of the computer. Moreover, in time-varying or non-stationary environments, it is important to ensure that stale information is unavailable during retrieval. Both of these requirements may be achieved via forgetting strategies (e.g., [30, 31, 32, 33]). For short-term memory systems (Level 1), this is achieved via the fixed-size working memory buffer that steadily marches past experience out of the buffer as new experiences come in. In Level 2-type systems that maintain session memory, forgetting occurs any time a new session is created. So, to guarantee proper maintenance of the memory system, a new session may be created whenever the system approaches the memory limit or when the environment dynamics change. Maintenance becomes a more pressing issue for levels of memory beyond 3. Here, forgetting may be implemented by case deletion (e.g., [33, 34]) or even by forgetting parts of cases and maintaining the rest [31]. Generalized or prototype cases may also enable the system to delete individual cases in preference of storing a single case that represents some average of the cases to delete, or only representing deviations from standard features [35].

For levels of memory at 3 and above, dealing with scalability is another challenge. At such levels, the memory will grow over time, potentially for decades. Despite this, it is essential that all the memory system operations (retrieval, adaptation, and case retention) execute efficiently. Consequently, the memory system needs an expressive, yet compact, representation that meaningfully describes cases, and supports long time-horizon operations. The system also needs to leverage clever indexing techniques to manage the organization of memory contents. Towards these ends, system designers might rely on storing and maintaining prototype or generalized cases as well as a hierarchical memory structure that supports efficient retrieval and retention.

A related but separate challenge to maintenance and scalability issues is relevance filtering. In memory-enabled LLM systems, past experience is exploited to enhance context and reasoning. The challenge lies in efficiently identifying and retrieving such useful information from episodic stores. A similarity metric heuristically guides the search for promising content. In a Level 1 or 2 system, all content can be searched and evaluated against a retrieval query. In higher-level memory systems, efficient retrieval strategies that make use of the organizational structure, or indexing scheme, of the memory are needed. Secondly, as far as the similarity metric is concerned, it must be designed to operate over the representations that are supported in the memory. For vector-based representations, this might be done using cosine similarity, embeddings, or other similarity measures that operate at the feature level. For structured content, similarity is determined via analogy, which is hard, computationally speaking. However, methods from CBR and analogical reasoning provide potential approaches (e.g., [36, 37]).

Lastly, because Level 4 and 5 require a general store of past experience, it is generally not possible for system designers to know ahead of time all the different possible ways a CBR-inspired memory system will exploit past experience to solve a given problem. In this situation, the traditional problem-solution pair case representation may not be appropriate. To overcome this, the memory system needs an adaptation technique that can dynamically set problem-solution pairs via partial matching against user input. Some CBR research has addressed problems for which problem and solution parts vary dynamically [38]. Another approach that could satisfy this is Bayesian inference over graph-based case representations. In recent work [39], cases are represented by Bayesian networks. The adaptation process dynamically maps

observed variables onto the problem part of the case, while variables to infer map onto the solution part.

6. Conclusions

Assessing the memory capabilities of AI systems in general—and of LLMs in particular—requires a standardized framework for delineating their capabilities. In response to this need, we have proposed such a framework, the Six Levels of AI Memory framework, capturing the major steps from simple memory-free systems, to short-term memory, session memory, long-term memory, human-like memory, and finally, superhuman memory. We have then presented a path, informed by this framework, for improving the memory capabilities of LLMs.

To form its vision for LLM memory, this paper revises and extends the opportunities and challenges related to CBR-inspired memory and LLM integration from the recent paper by Bach et al. [5]. CBR has a rich history related to the underpinnings of human memory and has developed methods for retrieving, storing, building and maintaining representation-rich memories, which form a promising basis for developing LLM memory systems. We encourage future investigations of memory and LLMs to build on the Six Levels of AI Memory framework and the opportunities and challenges presented here.

References

[1] K. Valmeekam, S. Sreedharan, M. Marquez, A. Olmo, S. Kambhampati, On the planning abilities of large language models (a critical investigation with a proposed benchmark), 2023. arXiv:2302.06706.

[2] J. Liu, A. Liu, X. Lu, S. Welleck, P. West, R. L. Bras, Y. Choi, H. Hajishirzi, Generated knowledge prompting for commonsense reasoning, arXiv preprint arXiv:2110.08387 (2021).

[3] B. Paranjape, J. Michael, M. Ghazvininejad, L. Zettlemoyer, H. Hajishirzi, Prompting contrastive explanations for commonsense reasoning tasks, arXiv preprint arXiv:2106.06823 (2021).

[4] M. Pink, Q. Wu, V. A. Vo, et al., Position: Episodic memory is the missing piece for long-term llm agents, 2025. URL: https://arxiv.org/abs/2502.06975. arXiv:2502.06975.

[5] K. Bach, R. Bergmann, F. Brand, M. Caro-Martínez, V. Eisenstadt, M. W. Floyd, L. Jayawardena, D. Leake, M. Lenz, L. Malburg, D. H. Ménager, M. Minor, B. Schack, I. Watson, K. Wilkerson, N. Wiratunga, Case-Based Reasoning Meets Large Language Models: A Research Manifesto For Open Challenges and Research Directions, 2025. URL: https://hal.science/hal-05006761.

[6] S. International, Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems , Technical Report, SAE Standard J3016, 2014.

[7] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in Neural Information Processing Systems 33 (2020) 9459–9474.

[8] R. C. Schank, Dynamic Memory: A Theory of Learning in Computers and People, Cambridge University Press, Cambridge, England, 1982.

[9] R. Schank, D. Leake, Natural language understanding: Models of Roger Schank and his students, in: Encyclopedia of Cognitive Science, volume 3, Nature Publishing Group, London, 2002, pp. 189–195.
[10] J. L. Kolodner, Reconstructive memory: A computer model*, Cognitive Science 7 (????) 281–328.
[11] M. Lebowitz, Integrated learning: Controlling explanation, Cognitive Science 10 (1986) 219–240.
[12] A. Kass, D. Leake, C. Owens, SWALE: A program that explains, in: Explanation Patterns: Understanding Mechanically and Creatively, Lawrence Erlbaum, Hillsdale, NJ, 1986, pp. 232–254.
[13] C. Riesbeck, R. Schank, Inside Case-Based Reasoning, Lawrence Erlbaum, Hillsdale, NJ, 1989.
[14] E. Domeshek, What Abby cares about, in: R. Bareiss (Ed.), Proceedings of the DARPA Case-Based Reasoning Workshop, DARPA, Morgan Kaufmann, San Mateo, 1991, pp. 13–24.
[15] D. Leake, An indexing vocabulary for case-based explanation, in: Proceedings of the Ninth National Conference on Artificial Intelligence, AAAI Press, Menlo Park, CA, 1991, pp. 10–15.
[16] R. Schank, R. Osgood, M. Brand, R. Burke, E. Domeshek, D. Edelson, W. Ferguson, M. Freed, M. Jona, B. Krulwich, E. Ohmayo, L. Pryor, A Content Theory of Memory Indexing, Technical Report 1, Institute for the Learning Sciences, Northwestern University, 1990.
[17] B. Peng, M. Galley, P. He, et al., Check Your Facts and Try Again: Improving Large Language Models with External Knowledge and Automated Feedback , CoRR abs/2302.12813 (2023). arXiv:2302.12813.
[18] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, Retrieval-augmented generation for large language models: A survey, arXiv preprint arXiv:2312.10997 (2023).
[19] K. Wilkerson, D. Leake, On Implementing Case-Based Reasoning with Large Language Models, in: Case-Based Reasoning Research and Development - 32nd International Conference, ICCBR 2024, Merida, Mexico, July 1-4, 2024, Proceedings, volume 14775 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 404–417.
[20] N. Wiratunga, R. Abeyratne, L. Jayawardena, et al., CBR-RAG: Case-Based Reasoning for Retrieval Augmented Generation in LLMs for Legal Question Answering , in: Case-Based Reasoning Research and Development - 32nd International Conference, ICCBR 2024, Merida, Mexico, July 1-4, 2024, Proceedings, volume 14775 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 445–460.
[21] C. Riesbeck, What next? The future of CBR in postmodern AI, in: D. Leake (Ed.), Case-Based Reasoning: Experiences, Lessons, and Future Directions, AAAI Press, Menlo Park, CA, 1996, pp. 371–388.
[22] D. Leake, CBR in context: The present and future, in: D. Leake (Ed.), Case-Based Reasoning: Experiences, Lessons, and Future Directions, AAAI Press, Menlo Park, CA, 1996, pp. 3–30. Https://homes.luddy.indiana.edu/leake/papers/p-96-01.pdf.
[23] K. Hammond, Case-Based Planning: Viewing Planning as a Memory Task, Academic Press, San Diego, 1989.
[24] S. Samsi, D. Zhao, J. McDonald, B. Li, A. Michaleas, M. Jones, W. Bergeron, J. Kepner, D. Tiwari, V. Gadepally, From words to watts: Benchmarking the energy costs of large lan-

guage model inference, in: 2023 IEEE High Performance Extreme Computing Conference (HPEC), 2023, pp. 1–9. doi:10.1109/HPEC58863.2023.10363447.
[25] D. Leake, R. Sooriamurthi, Case dispatching versus case-base merging: When MCBR matters, International Journal of Artificial Intelligence Tools 13 (2004) 237–254.
[26] D. Leake, D. Wilson, Categorizing case-base maintenance: Dimensions and directions, in: P. Cunningham, B. Smyth, M. Keane (Eds.), Proceedings of the Fourth European Workshop on Case-Based Reasoning, Springer Verlag, Berlin, 1998, pp. 196–207.
[27] E. Plaza, L. McGinty, Distributed case-based reasoning, Knowledge Engineering Review 20 (2005) 315–320.
[28] A. Goderis, P. Li, C. Goble, Workflow discovery: the problem, a case study from e-science and a graph-based solution, ICWS 0 (2006) 312–319. doi:http://doi.ieeecomputersociety.org/10.1109/ICWS.2006.147.
[29] P. Avesani, C. Hayes, M. Cova, Language games: Solving the vocabulary problem in multi-case-base reasoning, in: H. Muñoz-Ávila, F. Ricci (Eds.), Case-Based Reasoning Research and Development, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 35–49.
[30] M. Cox, A. Ram, An explicit representation of forgetting, in: Proceedings of the Sixth International Conference on Systems Research, Informatics and Cybernetics, Morgan Kaufmann, Baden-Baden, Germany, 1992.
[31] D. Leake, B. Schack, Flexible feature deletion: Compacting case bases by selectively compressing case contents, in: Case-Based Reasoning Research and Development, ICCBR 2015, Springer, Berlin, 2015, pp. 212–227.
[32] M. Salamó, M. López-Sánchez, Adaptive case-based reasoning using retention and forgetting strategies, Know.-Based Syst. 24 (2011) 230–247. doi:10.1016/j.knosys.2010.08.003.
[33] B. Smyth, M. Keane, Remembering to forget: A competence-preserving case deletion policy for case-based reasoning systems, in: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann, San Mateo, 1995, pp. 377–382.
[34] B. Smyth, E. McKenna, Competence models and the maintenance problem, Computational Intelligence 17 (2001) 235–249.
[35] M. Lebowitz, Generalization and Memory in an Integrated Understanding System, Ph.D. thesis, Yale University, 1980. Computer Science Department Technical Report 186.
[36] R. Bergmann, Y. Gil, Similarity assessment and efficient retrieval of semantic workflows, Information Systems 40 (2014) 115–127. URL: https://www.sciencedirect.com/science/article/pii/S0306437912001020. doi:https://doi.org/10.1016/j.is.2012.07.005.
[37] D. Gentner, K. Forbus, MAC/FAC: A model of similarity-based retrieval, in: Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society, Cognitive Science Society, Chicago, IL, 1991, pp. 504–509.
[38] D. Leake, A. Maguitman, T. Reichherzer, Experience-based support for human-centered knowledge modeling, Knowledge-based systems 68 (2014) 77–87.
[39] D. H. Ménager, D. Choi, S. K. Robins, A hybrid theory of event memory, Minds and Machines (2021) 1–30.