UC Merced
Proceedings of the Annual Meeting of the Cognitive Science Society

Title

Machines that Forget: Learning from retrieval failure of mis-indexed explanations

Permalink

https://escholarship.org/uc/item/1pn91687

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 16(0)

Authors

Cox, Michael T.
Cox, Michael T.
Ram, Ashwin

Publication Date

1994

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Machines that Forget: Learning from retrieval failure of mis-indexed explanations

Michael T. Cox

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280
cox@cc.gatech.edu

Abstract

A reasoner may fail at a cognitive task, not because it does not have appropriate knowledge with which to reason, but instead because it does not have the proper index or cue with which to retrieve such knowledge from memory. The reasoner knows this memory item; it simply cannot remember the item. This paper argues that forgetting provides an opportunity for learning through memory reorganization. A reasoner that takes full advantage of such opportunities, however, must be able to reason about its own memory system. To do so, it must possess a language for declaratively representing its reasoning failures and must reflectively inspect such representations if it is to fully explain the reason for its failure. Once such an error is understood as a memory failure, the problem of forgetting is to re-adjust the indexes so that the knowledge is properly retrieved in similar, future situations.

Introduction

The phrase "machines that forget" appears to be a contradiction in terms. Computer memory is often viewed as a virtually error-free medium in which retrieval of data is performed by simple fetch operations. As computer memories grow, however, brute-force search for the address to perform the fetch becomes increasingly intractable. Memory indexing is added in order to make memory retrieval more efficient. A memory-indexing mechanism is a trade-off between time to search and accuracy of retrieval; though efficiency is gained, poor indexing schemes risk not finding the proper information. That is, given some query, a computer may not find an item at all — from the user's point of view, it can "forget." Cognitive science research provides insights into this problem that take it beyond a mere computer science technicality. I will argue that development of improved computer memories for intelligent systems does not necessarily entail a memory that never forgets; rather, a useful memory is one that is integrated with a system that transforms forgetting into an opportunity to learn.

The *indexing problem* (Domeshek, 1992; Kolodner, 1984, 1993; Owens, 1993; Schank, 1982; Schank & Osgood, 1990) is that of choosing cues, or features in an input, to be used as indexes for retrieving from memory the knowledge structures necessary to process an input. The converse problem, then, is the *problem of forgetting* (Cox & Ram, 1992). If the cues are not chosen with care during retrieval time, or if the indexes are not chosen well during encoding, the reasoner may not recall a memory structure when it is needed. Thus, reasoning failures can occur because of faulty memory organization, as well as because of faulty reasoning components or faulty knowledge; forgetting is not simply a problem of "deleted" memory items. The solution to the problem of forgetting is to provide a system with an ability to recognize when its memory fails and the ability to associate these failures with abstract representations of the problem. It can then reason about the representations and associate them with specific learning goals. Performance systems that do not depend on brute-force search methods for information needs must therefore be integrated with learning systems that are sensitive to memory reorganization as well as to knowledge refinement.

Section 2 describes the theory and methodology that enables a system to reason about its own memory. Section 3 illustrates a solution to the problem of forgetting with an example from an implementation called Meta-AQUA and shows how it affects learning. Section 4 compares and contrasts both computational and psychological perspectives on the phenomenon of forgetting. Section 5 closes the paper with a brief discussion.

Introspective Multistrategy Learning

This paper illustrates how forgetting effects learning in a multistrategy learning system called Meta-AQUA (Ram & Cox, 1994). The system learns by choosing a learning strategy on the basis of introspective explanations of its own performance failures. The performance task for Meta-AQUA is story understanding. That is, given a stream of concepts as the representation for a story sequence, the task is to create a causally connected conceptual interpretation of the story. If the system fails at the task, its subsequent learning tasks are (1) *blame assignment* — analyze the cause of its misunderstanding, (2) *decide what to learn* — form a set of explicit learning goals to change its knowledge so that such a misunderstanding is not repeated on similar stories, and then (3) *strategy selection* — choose or construct some learning method by which it achieves these goals. The solution to these learning problems is to maintain a declarative trace of reasoning that leads to or supports a particular choice of goals or plans, to retrieve past cases of meta-reasoning that can explain the reasoning failure, and then to directly inspect and manipulate such explanations. The system's analysis of its failure is then used as a basis for generating specific learning goals and subsequently planning to achieve such goals by choosing a proper learning strategy.

An extension of explanation pattern (XP) theory (Ram, 1991, 1993; Schank, 1986) helps the system to reason about these types of failures. A *meta-explanation pattern* (Meta-XP) is an explanation of how and why an ordinary explanation fails in a reasoning system (Ram & Cox, 1994). Two classes of Meta-XPs exist to facilitate a system's ability to reason about itself and to assist in selecting a learning algorithm or strategy. A *Trace Meta-XP* (TMXP) explains how a system generates an explanation about the world or itself, and an *Introspective Meta-XP* (IMXP) explains why the reasoning captured in a TMXP goes awry. A TMXP records the structure of reasoning tasks and the reasons for decisions taken during processing in a series of decide-compute nodes. An IMXP is a general causal structure composed of primitive, network structures that represent various failure types. IMXPs are retrieved and applied to instances of reasoning captured in TMXPs, and assist in forming the learning goals of the systems after failure occurs. The algorithm that uses such knowledge structures is outlined in figure 1.

0. Perform and Record Reasoning in TMXP
1. Failure Detection on Reasoning Trace
2. If Failure Then
    - Learn from Mistake:
        - 2a. Blame Assignment
            - Compute index as characterization of failure
            - Retrieve Introspective Meta-XP
            - Apply IMXP to trace of reasoning in TMXP
            - If XP application is successful then
                - Check XP antecedents
                - If one or more nodes not believed then
                    - Introspective questioning
                    - GOTO step 0
            - Else GOTO step 0
        - 2 b. Create Learning Goals
            - Compute tentative goal priorities
        - 2 c. Choose Learning Algorithm(s)
            - Expand subgoals
            - Build learning plan
            - Compute data dependencies
            - Order plans
        - 2 d. Apply Learning Algorithm(s)
3. Evaluate Learning (not implemented)

Figure 1: Introspective Learning Algorithm

Once a system has identified the causes of a given reasoning failure (step 2a of figure 1), it must decide what it needs to learn. To represent these desires explicitly, it posts a series of learning goals that, if achieved, will reduce the likelihood of repeating the failure (step 2b). Some learning goals seek to add, delete, generalize or specialize some concept or procedure, or to reconcile or differentiate two concepts (Cox & Ram, 1994). Others deal with the ontology of the knowledge, that is, with the kinds of categories that constitute particular concepts.

Given a learning goal, then, a system must also decide which learning strategy is most appropriate for achieving it. Meta-AQUA treats the learning task like a traditional planning problem, creating a learning plan that is composed of a series of executions of learning algorithms that will achieve its learning goals (step 2c). However, unlike learning algorithms executed by single-strategy systems, the learner must dynamically consider possible interactions that may occur between the learning strategies (Cox & Ram, 1994). A non-linear planner is thus used to resolve learning-strategy dependencies and learning-goal interactions.

In this paper, we present extensions to Meta-AQUA that allow it to reason about memory failures. When reasoning about a failure such as forgetting, the system must first have a representation for the processes that preceded the failure. Having such a representation helps the system reason about itself, just as having declarative structures about events in the world assists problem solvers in reasoning about their environment. As will be seen in the following example, using these knowledge structures allows Meta-AQUA to pose questions about its own self-understanding.

Forgetting an Old Explanation

This section demonstrates how Meta-AQUA handles a failure in which it cannot generate an explanation for an anomaly. The explanation is in its memory, but the system does not have the proper index with which to retrieve it; in effect, it forgets the explanation. Given a bias for failure, this example demonstrates that forgetting represents an opportunity to learn.

Consider a story in which a police dog barks at luggage in an airport. This event is anomalous if the system believes that dogs bark only at animate objects. Meta-AQUA eventually learns that dogs can bark at any physical object, including inanimate ones, and it learns a new explanation: dogs bark when detecting contraband.

After processing this story, Meta-AQUA's memory contains knowledge representing two explanations for why dogs bark: an explanation for dogs that bark because they are threatened (indexed by dog-barks-at-animate-object) as well as an explanation for dogs that bark because they detect contraband (indexed by dog-barks-at-container).

Meta-AQUA is then given the following new story.

S1: The police officer and his dog enter a suspect's house.
S2: The dog barks at a pile of dirty clothes.
S3: The police officer looks under the clothes.
S4: He confiscates a large bag of marijuana.
S5: The dog is praised for barking at the occluding object.

The sentence, S1, causes no unusual processing because Meta-AQUA finds it mundane. But S2 is interesting because the system has recently changed its concept of `dog-bark`. The system therefore poses a question asking why the dog barked. Unfortunately, because the dog is barking at neither an animate object nor a container, no XP is retrieved with

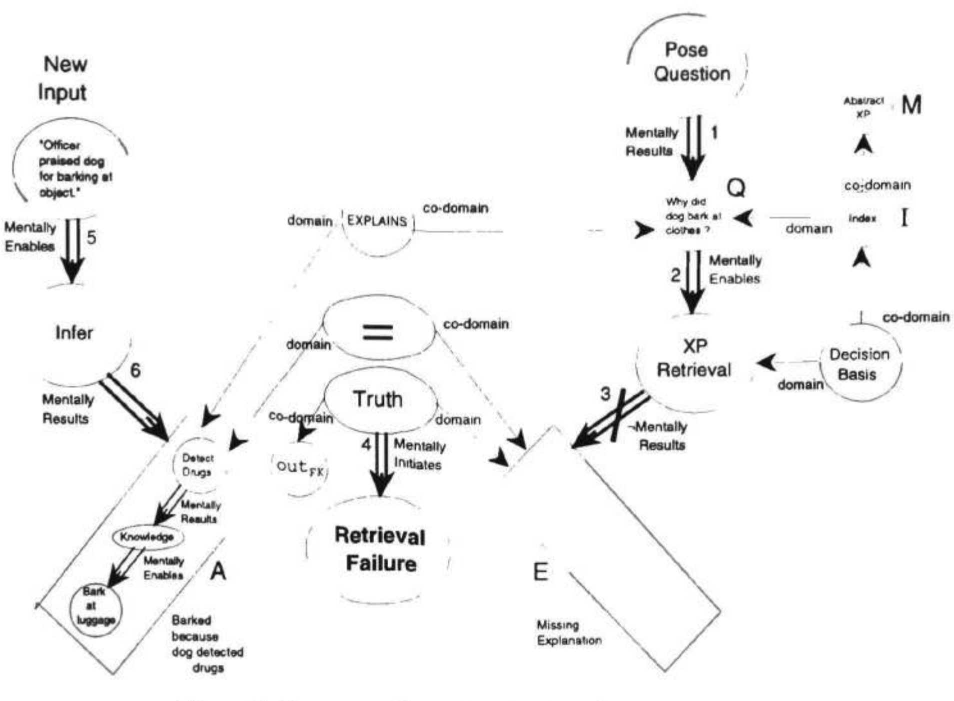

Figure 2: Forgotten Detection Explanation

which to generate an answer. The question-answering process is subsequently suspended because of the impasse, and the question is indexed in memory in the hope that the story will provide further information.

Next, sentence S3 causes the system to postulate a possible causal link between S2 and S3 simply because of the temporal relation; however, no evidence directly supports this hypothesis. S4 then reminds the system of a case in which contraband was confiscated. The system thus infers that the suspect was probably arrested. Finally, S5 causes a reminding of the earlier question about the dog barking at the pile of laundry. The reasoning that was associated with this previous question is then resumed. The system also infers a possible causal relation from S5. Although the sentence does not explicitly assert it, Meta-AQUA concludes that the dog's detection of the marijuana caused it to bark in the first place. This conclusion answers the original query.

Reviewing the reasoning trace that preceded the conclusion, Meta-AQUA characterizes itself as "baffled" (impasse during memory retrieval). The system retrieves an IMXP based on this characterization, which helps it explain its own reasoning failure.[1] The structure is unified with the representation of the original reasoning (stored in a TMXP) which produces the instantiation partially shown in Figure 2.[2] The knowledge structure shows that memory retrieval produced no explanation in response to the system's question. Instead, a later input caused the system to infer an answer.

The IMXP suggests that a knowledge-expansion goal be spawned to generalize the inferred explanation. This suggestion comes from a potential learning-goal slot of the IMXP (see Figure 3). Conditions attached to the knowledge-expansion goal allow it to be posted if the node A was either acquired from the story or inferred, but not if it was retrieved from memory. A knowledge-organization goal is also spawned in order to index the generalized explanation in memory. These goals can be achieved by performing explanation-based generalization (EBG) on the new explanation (node A) and then indexing the explanation by the context in which the system encountered the explanation.

The system cannot determine *a priori* whether an abstract XP (node M) actually exists in memory but could not be recalled (thus, the failure cause is a missing association, I), or whether the system lacks the knowledge to produce the explanation (thus, the cause is that the situation is novel, i.e., M is missing). The system thus poses a question about its own IMXP (c.f., Oehlmann, Edwards & Sleeman, 1994), "Does M exist in memory?" If M is missing, I is also missing; thus, the right question to ask is whether M exists, not I. Note that it cannot be the case that I is erroneous. If it were true, then some explanation would have been retrieved, although it may have been inappropriate.

The answer is obtained by performing EBG and then watching for a similar explanation in memory when it stores

1. The retrieved IMXP is called `IMXP-BAFFLED-AND-RESOLVED` and represents an instance of not remembering an explanation, yet later deriving one. An edited frame definition is shown in Fig. 3. Equal signs represent frame variable bindings.

2. Attributes and relations are represented explicitly in this figure. For example, the ACTOR attribute of an event X with value Y is equivalent to the relation `ACTOR` having `domain` X and `co-domain` Y. In addition, references to `TRUTH` attributes equal to `out` refer to the `domain` being out of the current set of beliefs. The subscript FK refers to it being out with respect to the foreground knowledge (as opposed to the background knowledge or BK). Numbers on the causal (double) links specify orderings.

the new explanation via the indexing algorithm. The system can detect the presence of similar memories by maintaining a list of pointers to memory items for each conceptual type. At storage time, Meta-AQUA traverses the list, checking each to see if it can unify the new memory with any of the older ones.[3] Meta-AQUA thus finds the explanation produced by the previous story at storage time.

```
(define-frame IMXP-BAFFLED-AND-RESOLVED
  (isa (composite-introspective-meta-xp))            ;; IMXP Class
  (failure-cause (novel-situation.0 missing-assoc.0)) ;; Which one we do not know
  (q (relation                                        ;; Baffling question
       (explanations (=a))))
  (a (xp                                              ;; Actual explanation
       (explains =q)))
  (e (xp                                              ;; Missing expectation
       (explains =q)))
  (i (index (domain =q) (co-domain =m)))              ;; Index used to retrieve E
  (m (xp))                                            ;; Forgotten xp.
  (truth-value (truth (domain =e)                     ;; E not in set of beliefs wrt FK
                      (co-domain out-fk.0)))
  (equals (equal-relation    (domain =a)              ;; Actual should have been
                             (co-domain =e)))         ;; equal to what was expected
  (rf (retrieval-failure     (initiates- =truth-value) ;; The memory failure
                             (expected-outcome =e)    ;; explained by the IMXP
                             (actual-outcome =a)))
  (new-input (entity))                                ;; Story input
  (later-process (process))                           ;; Inference in this case
  (rc (trace-meta-xp                                  ;; Reasoning chain
        (identification =q-id)
        (generation =hypo-gen)
        (link3 =link2)
        (link4 (mentally-results (truth out-fk.0)))))
  (q-id (d-c-node                                     ;; Question identification
        (strategy-choice questioning.0)
        (strategy-execution pose-question.0)
        (side-effect (considerations =con (prime-state =k-goal)))
        (link4 =link1)))
  (k-goal (knowledge-acquisition-goal                 ;; Knowledge goal to answer
            (goal-object                              ;; the question
              (generate (co-domain =q)))))
  (hypo-gen (d-c-node                                 ;; Hypothesis generation
              (strategy-decision =h-decision)
              (main-result (outcome =o (members (=link4))))
              (link4 (mentally-results (co-domain =o)
                                       (truth out-fk.0)))))
  (h-decision (decision-process                       ;; XP retrieval in this case
                (basis-of-decision =h-decision-basis)))
  (h-decision-basis
    (basis (knowledge                                 ;; Existence of I is the basis
             (collection                              ;; to use case-based explanation
               (members ((knowledge-state
                           (co-domain =i)
                           (believed-item =i)))))))))
  (links (=link1 =link2 =link3 =link4 =link5 =link6));; Links are in temporal order
  (link1 (mentally-results   (domain pose-question.0)
                             (co-domain (outcome (members (=q))))))
  (link2 (mentally-enables   (domain =con)
                             (co-domain =hypo-gen)))
  (link3 (mentally-results   (domain =rc)             ;; and all correspond to the
                             (co-domain =e)))         ;; numbered links in Fig. 2
  (link4 (mentally-initiates (domain =truth-value)
                             (co-domain =rf)))
  (link5 (mentally-enables   (domain =new-input)
                             (co-domain =later-process)))
  (link6 (mentally-results   (domain =later-process)
                             (co-domain =a)))
  (explains =rf)                                      ;; What the IMXP explains.
  (pre-xp-nodes (=a =e =rf))                          ;; XP consequents.
  (internal-nodes (=q =hypo-gen =later-process =i))   ;; Neither sink nor source nodes
  (xp-asserted-nodes (=q-id =m =new-input))           ;; XP antecedents.
  (potential-faults (=a =i))                          ;; Nodes for blame-assignment
  (potential-learning-goals                           ;; Corresponding learning goals
        ((knowledge-expansion-goal
          (goal-object =a)                            ;; Expand the new explanation
          (subgoals    =krg)
          (priority    (integer-value =pr))
          (backptr     (plan))
          (conditions  ((inferred.0 acquired.0)))
         (knowledge-reorganization-goal =krg
          (goal-object =i)                            ;; Reorganize memory to hold it
          (priority    (integer-value (less-than =pr)))))))))
```

Figure 3: IMXP Frame Definition

Merging the two explanations produces a better explanation: Dogs may bark at objects that hide contraband, not just at containers that hold contraband. The algorithm that indexes the generalization searches for the common ancestor of the object slots of both explanations; that is, objects that contain other objects and objects that cover other objects. This common ancestor is the type `hiding-place`. Thus, so that these types of explanations will not be forgotten again, the system indexes the explanation by "dogs that bark at potential hiding places" and places a pointer to the merged explanation on the memory list for the symbol `causal-relation`.

As a result of its learning, Meta-AQUA not only detects no anomalies in the following story, but predicts the correct explanation.

S1: A person is outside a house.
S2: The policeman approaches the suspect.
S3: His dog follows.
S4: The policeman sees that the person is near a compost pile.
S5: The dog barks at the compost pile.
S6: The authorities arrest the suspect for drug possession.
S7: The dog barked because he detected drugs.

Computational and Psychological Explanations of Forgetting

Early psychological theories of forgetting were based on the notion of the decay of a memory trace (e.g., Ebbinghaus, 1885/1964).[4] Many computational systems that model forgetting use a simple notion of decay or memory amortization to grossly simulate the phenomenon. Neural-net models of memory[5] implement forgetting by decreasing the weights on the connections between nodes in the system (for example, see Scalettar & Zee, 1988, for an explicit discussion of for-

3. This mechanism simulates a memory such as that of DMAP (Martin, 1990), whereby memory items map to areas that contain similar memories. Although Meta-AQUA's mechanism is only a crude approximation to such architectures, the emphasis of Meta-XP theory is on the reasoning about memory (or other reasoning processes), rather than on a representation of the memory architecture *per se*. A more realistic mechanism would be for Meta-AQUA to use the generalized XP as a probe to memory to see if it is now reminded of the old XP. The current method suffers from the fact that it always finds the old XP at an unacceptable search cost.
4. The most notable contemporary decay-theory of forgetting is Wickelgren's (1974) single-trace fragility theory of memory. The theory has been influential, not only in the psychological community, but also, for example, has been an inspiration for the forgetting mechanism in the user model of EUROHELP (Winkels, 1990), an intelligent help system.

getting in neural nets). However, forgetting itself is not a process; rather, it is a by-product of computation. The decay theory of forgetting is descriptive rather than explanatory.

Jenkins and Dallenbach (1924) introduced the concept of interference. Because learned information conflicts with similar memories, retrieval of these similar memories is interfered with. Forgetting is viewed as a competition between similar memories trying to associate with the current cue context[6]. The general search of associative memory (SAM) theory (Mensink & Raaijmakers, 1988) represents a current psychological model of forgetting and interference. Forgetting is represented as a lower probability of retrieving an item at time t+i than at time t. The SAM model assumes that the strength of a context cue during the retrieval of a particular memory item is determined by the overlap between the contexts at storage and retrieval times. Two factors explain forgetting in SAM. First, the cues used at time t may be more strongly associated with the memory item than those used at t+i. Second, the strength and number of competing items associated with the cues may be greater at t+i than at t. Although probability models cover the data, they do not explain the phenomena well; rather, like the notion of decay, they describe the phenomena. Moreover, the role of knowledge in memory retrieval is absent in such models.

Using a knowledge-intensive approach, CYRUS (Kolodner, 1984) models a dynamic memory (Schank, 1982) containing experiences from past US Secretary of State Cyrus Vance. Its memory contains knowledge of meetings concerning both the SALT Accords and Egypt-Israeli peace talks. But as CYRUS experiences more meetings between Egyptian and Israeli diplomats, the features that apply to the SALT talks are removed from the norms of the concept `diplomatic-meetings`. The SALT details are still in memory and retrievable if provided SALT-specific cues; however, the information concerning the Camp David Accords come to dominate the restructured concept. So when CYRUS is questioned about general diplomatic meetings, it forgets about SALT because of interference from the Camp David details. But CYRUS does not learn from its memory failures, nor does it have a representation of forgetting from which it can reason. As shown by the example in the previous section, Meta-AQUA learns to adjust its memory indexes in response to forgetting.

Rather than characterizing forgetting as to whether it results from interference or decay, a more useful strategy is to analyze the possible types of forgetting. Cox and Ram (1992) argue that the forgetting results from failure at various nodes in the structural representation of forgetting (as in Figure 1). As shown, the index, I, could be missing or the item, M, in memory could be missing. Alternatively, the system may never have generated the question, Q, in the first place. When confronted with later information, a system might realize in hindsight that it should have asked a question; therefore, Meta-AQUA can be surprised. Finally, an opportunistic reasoner might form a goal, but, because it cannot achieve that goal, may suspend the processing. Later it might forget to resume the goal. For example, if a planner forms the goal to fill its car with gas before leaving for vacation, the planner suspends the goal until arriving at the gas station. If the planner also has a goal to purchase supplies, then while at the gas station it may buy the supplies but forget to fill up the tank of the car. It is reminded of the forgotten goal when it runs out of gas while driving to its destination.[7] The failure occurred because it did not index the suspended goal with the features that would match contexts in which it would be appropriate to achieve the goal.[8] At this level, reasoning about forgetting appears useful, both for theorists and the reasoner itself.

Conclusion

Although not a formal memory model, the treatment of forgetting in this paper has shown that forgetting can be turned into an opportunity to learn. The lessons from the psychological literature suggest that to reason about forgetting it is important to consider the conditions upon which memories are retrieved (i.e., the relationship between the indexes used to store a memory and the cues available at retrieval time), rather than about decay.

Moreover, forgetting may be useful in additional ways, instead of being treated as a simple failure. Markovitch and Scott (1988) demonstrate that forgetting (as random deletions of memory items) may be useful in speeding up performance systems that use brute-force search techniques. Tambe, Newell, & Rosenbloom (1988) show how systems can decide whether or not to store an item based on the expected utility of the concept. Thus, an item in memory may be missing because it was never stored, or because it was stored and later deleted. However, a performance tradeoff exists between leaving an item in memory in the hope that it may be useful later and removing it to reduce storage space and speedup searching, with the additional cost of

5. A symbolic (as opposed to sub-symbolic) memory system that uses decay is ACT* (Anderson, 1983). Although ACT* uses a decay mechanism, Anderson has shown that ACT* covers much of the interference data in the literature. See the next paragraph for the interference explanation of forgetting.

6. Although Levy (1988) contends that neural nets model forgetting as interference, it is unclear from his discussion how they implement this. Neural nets use decay explanations as well as decay implementations. However, French (1994) discusses a more analytic evaluation of catastrophic interference phenomena in neural nets, whereby the addition of new memories in a net causes the system to completely forget association of older similar memories.

7. Suspended questions (knowledge goals) may also fail to be retrieved (e.g., forgetting to ask a question at the end of a lecture). Such a scenario is similar to forgetting to get gas.

8. Another explanation is that the reasoner might not have attended to the proper stimuli when at the store.

computing some criteria such as expected utility.

One of the problems with this paper's treatment of forgetting is that it concentrates on the encoding (storage) side of forgetting. Yet, Mensink and Raaijmakers (1988) assert that contemporary memory theories hold that forgetting is a retrieval effect, rather than an encoding effect. Deliberate elaboration and focus of attention during learning do control the storage context, and therefore affect encoding, so the storage issues are nonetheless important in any theory. More must be addressed in our theory, however, for effects of cue availability and selection during retrieval.

The solution to the problem of forgetting is to provide systems with the capability to notice when a reasoning failure is due to their memories along with the ability to associate these failures with abstract patterns that represent the problem. Systems can then reason about the representations and associate them with specific learning goals. When encountering memories with poor indexing, systems should reorganize the memory by learning better indexes; in response to surprises, systems need to learn when to ask the right questions; and when systems forget to remember their previous goals, they should learn to associate the goals with the right circumstances or learn to pay attention to the right cues in their environment. Like the stranded vacationer who forgot to fill up with gas and in response develops a habit of checking the gas gauge before going on long trips, we want computers to develop the right habits in their own domains.

Acknowledgments

AFOSR supported this research under grant number C-36-X26. I thank Ashwin Ram for his insight into this paper. I also thank Janet Kolodner and the anonymous reviewers for their feedback and Sue Farrell for proofing an earlier draft.

References

Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard Univ. Press.

Cox, M. T., & Ram, A. (1992). An explicit representation of forgetting. In J. Brahan and G. Lasker (Eds.), *Proc. of the 6th Intl. Conf. on Systems Res., Informatics and Cybernetics: Vol. 2.* (pp. 115-120). Windsor, Ontario: The Intl. Inst. for Adv. Studies in Systems Res. and Cybernetics.

Cox, M. T., & Ram, A. (1994). Choosing learning strategies to achieve learning goals. In M. desJardins & A. Ram (Eds.), *Proc. of the AAAI Spring Symp. on Goal-Driven Learning* (pp.12-21), Menlo Park, CA: AAAI Press.

Domeshek, E. A. (1992). *Do the right thing: A component theory for indexing stories as social advice* (Tech. Rep. No. 26). Doctoral dissertation, Evanston, IL: Northwestern Univ., Inst. for the Learning Sciences.

Ebbinghaus, H. (1964). *Memory: A contribution to experimental psychology.* (H. A. Ruger & C. E. Bussenius, Trans.) New York: Dover. (Org. pub. in 1885.)

French, R. M. (1994). Dynamically constraining connectionist networks to produce distributed, orthogonal representations to reduce catastrophic interference. In *Proc. of the 16th Annual Conf. of the Cog.Sci. Soc.*. Hillsdale, NJ: LEA.

Jenkins, J. B., & Dallenbach, K. M. (1924). Oblivescence during sleep and waking. *Am. J. of Psychology, 35*, 605-612.

Kolodner, J. L. (1984). *Retrieval and organizational strategies in conceptual memory: A computer model.* Hillsdale, NJ: LEA.

Kolodner, J. L. (1993). *Case-based reasoning*. San Mateo, CA: Morgan Kaufmann Publishers.

Levy, J. (1988). Computers that learn to forget. *New Scientist, 119*, 36-39.

Markovitch, S. & Scott, P. (1988). The role of forgetting in learning. In *Machine Learning: Proc. of the Fifth Intl. Conf.* (pp. 459-465). San Mateo: Morgan Kaufmann.

Martin, C. E. (1990). *Direct memory access parsing* (Tech. rep. 93-07). Doctoral dissertation, Chicago: University of Chicago, Dept. of Computer Science.

Mensink, G., & Raaijmakers, J. G. (1988). A model for interference and forgetting. *Psychological Review, 95*, 434-455.

Oehlmann, R., Edwards, P., and Sleeman, D. (1994). Changing the viewpoint: Re-indexing by introspective questioning. In *Proc. of the 16th Annual Conf. of the Cognitive Science Society.* Hillsdale, NJ: LEA.

Owens, C. (1993). Integrating feature extraction and memory search. *Machine Learning, 10* (3), 311-339.

Ram, A. (1991). A theory of questions and question asking. *Journal of the Learning Sciences, 1*(3&4), 273-318.

Ram, A. (1993). Indexing, elaboration and refinement: Incremental learning of explanatory cases. *Machine Learning, 10*(3), 201-248.

Ram, A., & Cox, M. T. (1994). Introspective reasoning using meta-explanations for multistrategy learning. In R. Michalski & G. Tecuci (Eds.), *Machine learning: A multistrategy approach IV* (pp. 349-377). San Mateo, CA: M. Kaufmann.

Scalettar, R., & Zee, A. (1988). Emergence of grandmother memory in feed forward networks: Learning with noise and forgetfulness. In D. Waltz & J. A. Feldman (Eds.), *Connectionist models and their implications* (p. 309-327). Norwood, NJ: Ablex.

Schank, R. C. (1982). *Dynamic memory: A theory of reminding and learning in computers and people*. Cambridge, MA: Cambridge Univ. Press.

Schank, R. C. (1986). *Explanation patterns: Understanding mechanically and creatively.* Hillsdale, NJ: LEA.

Schank, R. C., and Osgood, R. (1990). *A content theory of memory indexing* (Tech. Rep. No. 2). Evanston, IL: Northwestern Univ., Inst. for the Learning Sciences.

Tambe, M, Newell, A., and Rosenbloom, P. (1990). The problem of expensive chunks and its solution by restricting expressiveness. *Machine Learning, 5*, 299-348.

Wickelgren, W. (1974). Single-trace fragility theory of memory dynamics. *Memory & Cognition, 2*, 775-780.

Winkels, R. (1990). User modelling in help systems. In D. H. Norrie & H. -W. Six (Eds.), *Proc. of the 3rd Intl. Conf. on Comp. Assis. Learning* (pp. 184-193). New York: Springer.