Organizing Memory for Explanation*

David B. Leake and Christopher C. Owens
Yale University

ABSTRACT

We present a mechanism for remembering explanations and re-using them to explain new episodes. This task requires a representation scheme for explanations, a dynamically organized memory, and a means of modifying old explanations to fit new facts. In this paper we focus on memory organization. We describe strategies for indexing and retrieving explanations, for using causal knowledge to select relevant features of episodes and for guiding generalization. We discuss work in progress on a computer implementation of this model.

INTRODUCTION

A task that people are able to perform, and one we have often required of Artificial Intelligence systems, is to make explanations. The ability to explain things has long been considered to be a yardstick for judging the depth to which a program has understood a situation. Yet recently we have begun to see explanation as more than just a task requiring understanding; we have begun to see explanation as an integral and necessary component of the understanding process itself [Schank 86].

In order to understand a situation, we must be able explain it to ourselves, that is, to connect it in a useful way with the rest of our knowledge. The type of explanation required for a given situation depends on the task which the explainer wishes to perform. For some AI programs, explanations relate facts to currently-active memory structures like scripts [Cullingford 78] or similar schemas guiding top-down processing. Such an approach explains a fact like "John left a tip" by, in effect, saying "because that is what typically happens in restaurant situations." Explanations can also relate actions to the goals they satisfy, as in the case of SHRDLU [Winograd 72], which could explain any of its actions by relating them back to the goals they satisfied. Explanation of less stereotyped information has required chaining together pieces of primitive causal knowledge into more elaborate explanatory chains. PAM [Wilensky 78] could relate an input fact to goals via known plans, for example explaining why a person would buy a gun if he intended to commit a robbery.

While building explanations by chaining together small pieces of causal knowledge increases flexibility, undirected backwards chaining causes a combinatoric explosion of connections to consider. Although not a problem in highly restricted domains, combinatoric explosion is rapidly aggravated by increasing knowledge. This violates our intuition that increasing knowledge and experience should facilitate building explanations, not make the process more difficult.

We (working jointly with Alex Kass) are developing a program to construct explanations of complicated real-world events, events for which people have no ready explanations. Making difficult explanations by building up connections from scratch would be overwhelming; we need to bring connections learned from relevant experience to bear on the process. To be able to retain these relevant experiences, we must have appropriate memory structures to store past explanations; for the information to be accessible, we must be able to form useful categories to organize them.

This paper describes a mechanism, implemented in our program, for categorizing explanations in

*This work is supported in part by the Air Force Office of Systems Research under contract 85-0343

memory, retrieving them where relevant, and for applying them to new situations.

NEW EXPLANATIONS FROM OLD

An episode for which we have been collecting anecdotal evidence is the death of the highly successful young racehorse Swale, who died one week after having won the prestigious Belmont Stakes horse race. Although few facts had appeared in the newspapers and although most people have little specialized knowledge to apply to explaining racehorse death, people were nevertheless able to quickly hypothesize a variety of plausible explanations.

One of these explanations was that Swale's death was like the ironic death of Jim Fixx, a prominent advocate of the health benefits of jogging. Fixx died while running, as a result of a congenital heart defect. According to this explanation Swale had a hidden heart problem that was exacerbated by the strain of his all-out effort to win the race.

Another explanation was that Swale had been killed by his owners to collect the insurance money. (This quickly turned out to be implausible when it was revealed that Swale had been under-insured.)

The questions raised by these explanations are: How did these people get reminded of Jim Fixx on the one hand and of the notion of insurance fraud on the other, how did they retrieve the explanations corresponding to these concepts, and how were they able to apply them to the circumstances of Swale's death?

Explanation and reminding are closely bound together. While [Schank 82] observed that one episode can remind people of another if the two have a thematically common explanation, we claim that the process can run in the other direction as well: that explanations can be constructed from a reminding. If, during the course of processing an episode, an understanding system can be reminded either of some other episode whose explanation is relevant or of an explanation template that has been used to explain similar situations in general, then the system can apply the old explanation to the task of understanding the new episode.

The knowledge structure we use for storing explanations is the Explanation Pattern (XP), initially outlined in [Schank 86]. XPs contain a template of the kind of episode they are designed to explain, with causal annotation identifying the connections between the features of the episode. When a near-miss explanation needs modification to fit the current situation, this causal structure is essential to the revision process. How problems with near-miss explanations are identified, and how the revision or "tweaking" of explanations is done, are beyond the scope of this paper; they are described in more detail in [Kass 86] and in [Kass, Leake and Owens 86].

Our explanation algorithm is as follows:

ANOMALY DETECTION Attempt to fit story into memory. If successful DONE; otherwise an anomaly has been detected.

XP SEARCH Search for an XP that can be applied to explain the anomaly.

XP ACCEPTING Attempt to apply XPs. If successful then skip to XP INTEGRATION.

XP TWEAKING If unable to apply XPs directly then attempt to tweak them into XPs that might apply better. If successful send the tweaked XPs back to XP ACCEPTING.

XP INTEGRATION If any results are accepted, integrate results back into memory making appropriate generalizations.

This paper focuses primarily on XP Search and XP Integration.

EXPLANATION MEMORY

A system that re-uses old explanations to understand new episodes must have a way of storing old explanations and a means of finding them when appropriate to help in understanding new situations. This memory must fail gracefully: if it can't find an explanation exactly suited to the current episode, the near-misses must serve as departure points for new explanation creation. Continuing the Swale example, explanation patterns about death of racehorses would obviously be relevant, as would explanation patterns about deaths of athletes, about deaths of the young and famous, about destructions of important income-producing properties, and about other bad things that have happened to racehorses. How can one find these explanation

patterns so that they might be proposed as candidates?

Although we agree with [Gentner and Landers 85] that access to relevant knowledge structures (in our case Explanation Patterns, in her case analogies) can be based upon surface features, we believe strongly that all features of an episode are not used equally in a search for applicable structures. An intelligent process must select the features to be used as indices into memory, and the search process must complement the processes by which explanation patterns have been indexed.

As an example of the latter kind of indexing process, here are some indexing strategies which can be used to decide how to store XPs in memory. We suspect that these indexing methods are a few among what will ultimately be many.

Indexing rule 1: Index an XP via features participating in the anomaly that the XP is designed to resolve. For example, Jim Fixx's death was surprising because his health seemed outstanding; the Jim Fixx XP described above can be indexed under the combination: (Death + Excellent Physical Condition)

Indexing rule 2: Index an XP via a feature playing a role in the chain of causation contained within the explanation. Using this strategy, the Jim Fixx XP could be indexed under: (Death + Heart Defect) or (Death + Jogging)

Indexing rule 3: Index an XP via any highly unusual feature of the episode that it originally explained, whether or not that feature played a causal role in the explanation. If you get a flat tire during a blizzard, another instance of car problems in blizzards may remind you to get your tires retreaded.

Indexing rule 4: Index an XP via features defining membership in a commonly-stereotyped group. (A group that has been previously defined for other purposes.) For example, one of the explanations we collected for Swale's death centered around other deaths of famous young star performers, with remindings of Jimi Hendrix and Janis Joplin. This explanation pattern is indexed under (Death + Successful Star Performer)

Once XPs have been indexed using the above methods, complementary retrieval strategies can be applied to new episodes in order to find relevant XPs. Following are some retrieval strategies associated with the above; again we expect this list to grow.

Retrieval Strategy 1: Consider directly-indexed XPs. We claim that for a given initial categorization of an event or for a given anomaly type, there are likely to be indexed a small number of immediate candidate explanation patterns addressing the situation. Under death, for example, we might find old age and sickness.

Retrievel Strategy 2: If an XP fails to fit, consider the features that have caused it to fail. If we try old age as an explanation of Swale's death, it fails because Swale was young. So, we look to see what we have indexed as explanations of early death. Similarly if we try death from sickness, we find a contradiction with Swale's excellent condition inferred from his recent racing victory. This leads us to examine explanation patterns indexed under death and excellent condition, which yields the Jim Fixx reminding.

Retrieval Strategy 3: Consider extreme or unusual features of the current episode. Swale, for example, was of great monetary value compared with the normative instance of a racehorse. Death and great monetary value can index a variety of obvious XPs, such as being killed for the insurance money.

GENERALIZING AND FORMING CATEGORIES

Within an XP memory, it's important to generalize explanation patterns into categories based on shared causal structure. When XPs are retrieved, near misses will be useful if causally-related XPs are stored under similar indices– conceptually near each other.

There initially seems to be a problem with circularity here: one reason for doing explanation is to select the relevant features for categorizing an episode, but these features are the ones needed in order to retrieve a correct XP. However, this apparent circularity is avoided by having static methods to initially select a set of candidate features for use as indices, even though these indices may retrieve irrelevant or wrong XPs. Once a set of XPs is available as a starting point, dynamic methods

modify them into final explanations embodying a reasonable set of features from the episode.

Early attempts to organize a memory full of knowledge structures used straightforward inductive category formation. An example of this approach was IPP [Lebowitz 80]. It read newspaper stories about terrorist attacks, using MOPs [Schank 82] to provide expectations during the story understanding process. By extracting features shared across multiple stories, the program formed generalized MOPs representing the features common to those episodes. These new categories could then be used in understanding future stories. This method is a realistic approximation to one kind of generalization people actually do. For example, from reading newspaper stories about Italy when the Red Brigades were active, IPP formed the generalization that the usual victims of kidnapping in Italy were businessmen.

PROBLEMS WITH INDUCTION

But systems using the type of induction described above have a number of problems (which are discussed further in [Schank, Collins and Hunter 86]). Inductive learning systems depended upon having multiple episodes available. Their idea was that, as many episodes were processed, features resulting from presumably randomly-distributed noise would tend to cancel each other out, leaving generalizations containing only important features. But looking only at frequency of feature appearance has two problems. One is that people can in fact form generalizations in one trial, which is not accounted for by a feature frequency method. The second is that, without any metric of feature importance, induction learning systems, when faced with small numbers of episodes, tend to form silly generalizations, for example: "Terrorist attacks in India always kill exactly two people." People form erroneous generalizations, but not of this variety.

Furthermore, the very notion of saving common features across multiple episodes depends on a representation scheme in which some features are clearly part of the episode and others are not. One does not want to take a complete description of the state of the world at the time of two separate episodes and generalize all the features that were true at both times. Instead, one wants to take two episode representations and generalize the features they have in common. Limiting the feature space this way is a reasonable strategy for certain domains, particularly technical reasoning where the number of factors is known and limited. But for general-purpose understanding, limiting the representation scheme to one in which each episode has only a few features results in an unnaturally sparse and impoverished knowledge base. A key component of the understanding task is focusing attention: deciding, among all the things that happen to be true of the world at a particular time, which of them are reasonably part of a given episode and which are not. A representation that tries to solve this a priori does not help much towards a psychologically interesting theory of understanding, nor does it have the potential to ultimately yield sophisticated automated reasoners.

USING CAUSAL LINKS

Within our model, the causal annotation of XPs is crucially important to the tasks of modifying and generalizing explanations. When we try to relate the Jim Fixx explanation to Swale's death, we don't simply consider every aspect in which Jim Fixx and Swale were similar. Instead, we use the causal structure of the Jim Fixx explanation to see that the relevant fact about Jim Fixx was that he did regular strenuous exercise.

In our search through aspects of Swale, we don't spend any search time considering, for example, that Jim Fixx was also an author or that joggers typically get up early in the morning. Those facts are present and could be accessed by some different XP, but because they are not causally connected with this particular view of Jim Fixx's death they do not participate in this attempt to explain Swale's death. However, if we tried to apply an XP about ironic deaths of famous people to Jim Fixx, then the fact that he wrote books advocating jogging would certainly be considered in any attempt to generalize his death with other events (such as the death of the famous natural foods advocate who reputedly died of stomach cancer.) XPs provide a specific, causally connected view of a class of events; trying to apply an XP to a new event tightly constrains the features that will be considered as candidates for inclusion in a description and explanation of the new event.

Once we have accepted a modified XP as the explanation of an event, we attempt to generalize the old

and revised XPs in order to form a category that subsumes both explanations. Causal links within the original explanations constrain the generalization process: any generalized condition must support the same causal chains as the conditions it subsumes. Our explanation of Fixx's death differs from that of Swale since it depended on the actor jogging, while Swale's depended on his running in horse races. To generalize these premises, we need to find not only an abstraction they share (such as both being participation in outdoor activities) but one that fits each causal pattern. Since physical exertion is the abstraction that satisfies this requirement, it is the generalization selected. Our new explanation is that an actor in apparently good physical condition may die if he does a lot of exertion and has a hidden heart defect.

Thus in our system, the formation of generalizations is driven by the need to accomodate new events in memory. There is a difference between the knowledge which is accessible to the system in the form of basic causal knowledge, and the more complex patterns which our program applies to new situations. Similarly we maintain the difference between knowledge that is simply present in the system and knowledge that is accessible under a particular index. Rather than trying to generalize an explanation pattern as far as possible without regard to whether the generalization is needed to deal with our domain, we generalize an XP only when it can no longer accomodate the data.

CONCLUSIONS

The mechanism we have described allows an understander to retrieve and apply relevant old explanations to new situations, and to use information from the explanations to guide its categorization of episodes in memory. This process accounts for certain kinds of concept formation in small numbers of trials, and avoids some of the faults of straight induction. It also avoids some of the difficulties faced by other explanation systems (e.g., [Segre and Dejong 85]) in that the understander is not required to build up an explanation from scratch each time a situation is encountered. Instead, our system can resolve new problems more efficiently as its library of explanation patterns grows with experience.

Our theory of explanation bears on focus of attention in a situation: we suggest some strategies for selecting features to use as indices when searching memory for relevant patterns. Even when these strategies yield no directly-applicable pattern, our system fails gracefully in that it can use a near-miss pattern as the starting point for building a new explanation. Our system is still in its preliminary stages; further work on explanation memory will involve refining and extending the strategies used to search for XPs and to evaluate whether a given XP is satisfactory.

ACKNOWLEDGEMENT

The work described in this paper was done jointly by the authors, Alex Kass, Roger Schank and Christopher Riesbeck, all at Yale. We would like to thank Chris Riesbeck for his helpful comments on drafts of this paper.

REFERENCES

[Cullingford 78] Cullingford, R., *Script Application: Computer Understanding of Newspaper Stories,* Ph.D. Thesis, Yale University, 1978. Research Report #116.

[Gentner and Landers 85] Gentner, D. and Landers, R., Analogical Reminding: A Good Match is Hard to Find, *Proceedings of the IEEE 1985 International Conference on Systems, Man and Cybernetics,* IEEE, 1985, pp. 607ff.

[Kass, Leake and Owens 86] Kass, A. M. and Leake, D. B. and Owens, C. C., *SWALE: A Program that Explains,* 1986. In [Schank 86].

[Kass 86] Kass, A. M., Modifying Explanations to Understand Stories, *Proceedings of the Eighth Annual Conference of the Cognitive Science Society,* Cognitive Science Society, Lawrence Erlbaum Associates, 1986.

[Lebowitz 80] Lebowitz, M., *Generalization and Memory in an Integrated Understanding System,* Ph.D. Thesis, Yale University, October 1980.

[Schank, Collins and Hunter 86] Schank, R.C. and Collins, G. and Hunter, L., *Transcending Inductive Category Formation in Learning,* Behavioral and Brain Sciences, 9/4 (1986). In Press.

[Schank 82] Schank, R.C., *Dynamic Memory: A Theory of Learning in Computers and People,* Cambridge University Press, 1982.

[Schank 86] Schank, R.C., *Explanation Patterns: Understanding Mechanically and Creatively,* 1986. Book in press.

[Segre and Dejong 85] Segre, A. M. and DeJong, J., Explanation-Based Manipulator Learning: Acquisition of Planning Ability through Observation, *Proceedings of the IEEE 1985 International Conference on Robots and Automation,* IEEE, 1985, pp. 555ff.

[Wilensky 78] Wilensky, R., *Understanding Goal-Based Stories,* Ph.D. Thesis, Yale University, 1978. Research Report #140.

[Winograd 72] Winograd, T., *Understanding Natural Language,* Academic Press, New York, 1972.