Explanation and Generalization Based Memory

Michael J. Pazzani
UCLA Artificial Intelligence Laboratory
3731 Boelter Hall
Los Angeles, CA 90024
and
The Aerospace Corporation
P.O. Box 92957
Los Angeles, CA 90009

Abstract

A model of memory and learning is presented which indexes a new event by those features which are relevant in explaining why the event occurred. As events are added to memory, generalizations are created which describe and explain similarities and differences between events. The memory is organized so that when an event is added, events with similar features are noticed. An explanation process attempts to explain the similar features. If an explanation is found, a generalized event is created to organize the similar events and the explanation is stored with the generalized event.

Introduction

The goal of this research is to identify the role of explanation in a generalization based memory. A computer program, OCCAM, has been implemented which learns about variations of kidnapping. The program starts out with general knowledge about coercion represented as a meta-MOP [7]. After some examples, it creates a MOP which describes a kind of kidnapping (along with the explanation that a family member of the victim's family pays the ransom to achieve the goal of preserving the victim's health). Further examples create a specialization of this MOP which represent an inherent flaw in kidnapping: that the victim can testify against the kidnapper, since the kidnapper can be seen by the victim. This specialization is stored as a sub-MOP of the kidnapping MOP and is indexed by the kidnapper's goal failure: going to jail. After some more examples, a similarity is noticed about the kidnapping of infants. This coincidence starts an explanation process which explains the choice of victim to avoid a possible goal failure, since infants cannot testify.

There are a couple of interesting features of this type of learning:

- The explanation process eliminates the problem of including unrelated coincidences in generalized events. For example, all of the infants kidnapped in the events presented to OCCAM have blond hair. This feature is not used in the explanation, so it is not included in the generalized event.

- There is causal and motivational information associated with generalized events. This information states why various features are included in the generalized event.

- The explanation process can make use of the generalized events in memory. Explanation consists of a rule based component similar to PAM [9] and a memory based explanation component. The rules state such things as that if someone says they are going to hurt a family member, this motivates a goal of preserving the health of the family member. There are no special rules about kidnapping. Therefore, it is not capable of explaining the kidnapping of infants until it has built a generalized event about the victim testifying against the kidnapper. The explanation process uses intentional links [2] to specify the relationships between goals, plans and events.

Related Work

Much early work on learning (e.g., [8], and [3]), centered on the acquisition of a concept from a number of examples. A characteristic description of a class of objects was built by inductive means by considering positive (and, in some instances, negative) examples. The work reported here differs from this work on concept acquisition in a number of ways. First, these programs were "told" what concepts to learn and examples were identified as positive or negative instances. In contrast, OCCAM is not told what to learn. Instead, OCCAM incrementally learns new concepts from examples as a natural consequence of organizing memory around similarities. Secondly, the generalized events built by OCCAM do not contain all features common to the examples. Its explanation process distinguishes between relevant and coincidental features.

DeJong presents a model of explanation based learning [1] which learns schemata from a single example. His program constructs an explanation of relationships between various components of an event by a knowledge-intensive understanding process similar to PAM. The explanation and the event are then generalized by retaining only those parts used in the explanation. Our work differs from DeJong's in a number of aspects. First, OCCAM learns incrementally. It is difficult to imagine a system learning the specialized motivation for kidnapping infants from the first example of a kidnapping, since the explanation process can find an explanation for kidnapping any person. In OCCAM, after the basic kidnapping schema (or MOP) is learned, later examples focus OCCAM on explaining coincidences about the age of the victims. Additionally, the explanation process makes use of other events or generalized events.

In IPP [5] and UNIMEM [6], Lebowitz is concerned with making "factual" generalizations. Unlike OCCAM, these programs make no attempt to perform a causal or explanatory analysis. Therefore, no distinction is made between relevant or coincidental features. After a number of diverse examples, IPP and UNIMEM can correct generalizations to remove coincidences which are contradicted.

CYRUS [4] is a program which organizes and searches a model of episodic memory. Like IPP, it does not produce an explanation of its generalizations. It avoids the problem of indexing on coincidentally similar features by an a priori set of relevant features.

Learning and Memory in OCCAM

OCCAM makes a distinction between two types of generalized events. *Explanatory generalized events* are MOPs created as a specialization of a more general MOP. Associated with each explanatory generalized event are new causal and goal relationships. For example, the kidnapping of infants is an explanatory generalized event which includes the special motivation for selecting the hostage. *Organizational generalized events* are also created as specialization of more general MOPs. However, they add no additional explanatory information. They correspond to the factual generalizations of IPP and serve mainly to organize the memory. An example organizational generalized event would be kidnappings where the hostages grandmother paid the ransom. (Unless, of course, some explanation could be found.)

There are two parts to the incremental learning algorithm used by OCCAM. The first step is to find the appropriate place in memory to index a new event. The memory is organized so that a new event will be added to memory in the same place as similar events. The second step is to attempt to create a generalization.

After the most specific applicable MOP is found, similar events are found by using the features of the new event as indices. Next, generalization is attempted by a number of *generalization rules* which postulate causal or intentional relationships. For example, one rule states *If an action always precedes a state, postulate the action causes the state.* Other rules which postulate goal relationships will be discussed in the next section. An explanation process is then used to verify the postulated causal or intentional relationships. This explanation process marks all features necessary for establishing the relationships. The explanation process used here is more focused than that used in DeJong's work. Rather than asking general questions such as "Why did this happen?", more specific questions are used such as "Was there an action which motivates a goal before this action which this action achieves." A

new MOP may be created depending on the result of the generalization:

- If the explanation is successful, than an explanatory generalized event is built and indexed under the most specialized MOP by the new features establishing the explanation. The new event and any similar events are organized under this new generalization, indexed by the features not used in the generalized event.

- If the explanation process is unsuccessful, and the most specific MOP is an explanatory generalized event, a default rule is used to attempt to form an organizational generalized event. This notes that there appears to be a coincidental relationship but does store any justification.

An Example: Learning about Kidnapping

The meta-MOP for coercion involves a PREParation, a THREAT, a DEMAND, and several RESULT scenes. Figure 1 illustrates an example of kidnapping which is a kind of coercion. In this Figure, the notation "the(FEATURE)" indicates the actual value of the feature is the same as the value of that feature. Coercion usually involves at least three roles: an ACTOR, who performs the PREParation, and says he will carry out the THREAT unless his DEMAND is met; an OBJECT which is the object of the PREParation and the THREAT (i.e., in kidnapping the hostage is the OBJECT); and the VICTIM which receives the THREAT, and usually performs one of the RESULTs. (The VICTIM in kidnapping is not the hostage but the person who pays the ransom.) The coercion meta-MOP is intended to be very general and account for many situations from kidnappings to playground arguments (e.g., "If you don't let me pitch, I'm gonna take my ball and go home").

```
-------------------------------------------------------------
K1: COERCION
ACTOR human  NAME Joe K.  HEIGHT tall AGE 30s HAIR brown
OBJECT human  NAME John V.  HEIGHT short AGE teens HAIR blond
              RELATION family TYPE son
                              OF the(VICTIM)
VICTIM human  NAME Dad V.  HEIGHT tall AGE 40s HAIR blond
              RELATION family TYPE father
                              OF the(OBJECT)
PREP atrans ACTOR the(ACTOR) TO the(ACTOR) OBJECT the(OBJECT)
DEMAND poss-by ACTOR the(ACTOR)
               OBJECT money AMOUNT 50000
THREAT health OF the(OBJECT) VAL -10
RESULT atrans  ACTOR the(ACTOR) FROM the(ACTOR)
               TO the(VICTIM) OBJECT the(OBJECT)
RESULT atrans ACTOR the(VICTIM)
              FROM the(VICTIM)
              TO the(ACTOR)
              OBJECT money AMOUNT 50000
RESULT $trial SENTENCE 15 VERDICT guilty
              WITNESS the(OBJECT) CRIMINAL the(ACTOR)
-------------------------------------------------------------
```

Figure 1: An Example of Coercion: A Kidnapping

The initial state of the memory of OCCAM contains only the coercion meta-MOP (mm-COERCE). K1, the example in Figure 1, is then added to memory. It is indexed under mm-COERCE by all of its features (i.e., its scenes and roles). The next example, K2, is similar to K1, except the AGE of the OBJECT is an infant, some minor difference in the features of the VICTIM and the ACTOR, and there is

```
Looking for similar events under mm-COERCE... found (K1).
Similarities:
COERCION
ACTOR human  HEIGHT tall AGE 30s HAIR brown
OBJECT human RELATION family TYPE son
                              OF the(VICTIM)
VICTIM human RELATION family TYPE father
                              OF the(OBJECT)
PREP atrans ACTOR the(ACTOR) TO the(ACTOR) OBJECT the(OBJECT)
DEMAND poss-by ACTOR the(ACTOR) OBJECT money
THREAT health OF the(OBJ) VAL -10
RESULT atrans ACTOR the(ACTOR) FROM the(ACTOR)
              TO the(VICTIM) OBJECT the(OBJECT)
RESULT atrans ACTOR the(VICTIM) FROM the(VICTIM)
              TO the(ACTOR) OBJECT money
-----------------------------------------------------------------
```

Figure 2: Noticing the Similarities between two kidnappings

no trial in which the ACTOR goes to jail. Figures 2 and 3 are an edited transcript of the creation of a MOP which describes the kidnapping of a family member for a monetary ransom. The similarities between K1 and K2 are noted (see Figure 2). Then, a rule, GENERALIZE-RESULTS, is used to postulate an explanation for this similarity. This rule states *Look for an action before the RESULT which motivates a goal which the RESULT achieves or an action before the RESULT which is part of plan which the RESULT realizes.* In this example, a goal of preserving the health of the OBJECT by the VICTIM is inferred and paying the ransom achieves this goal. Additionally, the ACTOR is performing the plan of keeping a bargain when he gives the OBJECT back. In general, a better explanation utilizes the goals rather than the plans. However, in this case, it's not possible to infer why the kidnapper releases the hostage. This new MOP (MOP.327) is indexed under mm-COERCE by the relevant features, and the inferred goal as shown in Figure 4. Notice that some features (e.g., the AGE, and HEIGHT of the kidnapper) are not included in the generalization even though they are common to both kidnapping examples because they are not used in the explanation.

```
-----------------------------------------------------------------
Running generalization rule GENERALIZE-RESULTS.
Inferring RESULT REALIZES PLAN (KEEP-BARGAIN)
Inferring RESULT ACHIEVES GOAL (P-HEALTH)
Making sub-mop MOP.327 {kidnap} of mm-COERCE from (K2 K1)
Used in explanation:
COERCION
ACTOR human
OBJECT human
VICTIM human RELATION family OF the(OBJECT)
DEMAND poss-by ACTOR the(ACTOR) OBJECT money
THREAT health OF the(OBJECT) VAL -10
RESULT atrans ACTOR the(ACTOR) FROM the(ACTOR)
              TO the(VICTIM) OBJECT the(OBJECT)
RESULT atrans ACTOR the(VICTIM) FROM the(VICTIM)
              TO the(ACTOR) OBJECT money
-----------------------------------------------------------------
```

Figure 3: Forming an Explanatory Generalized Event

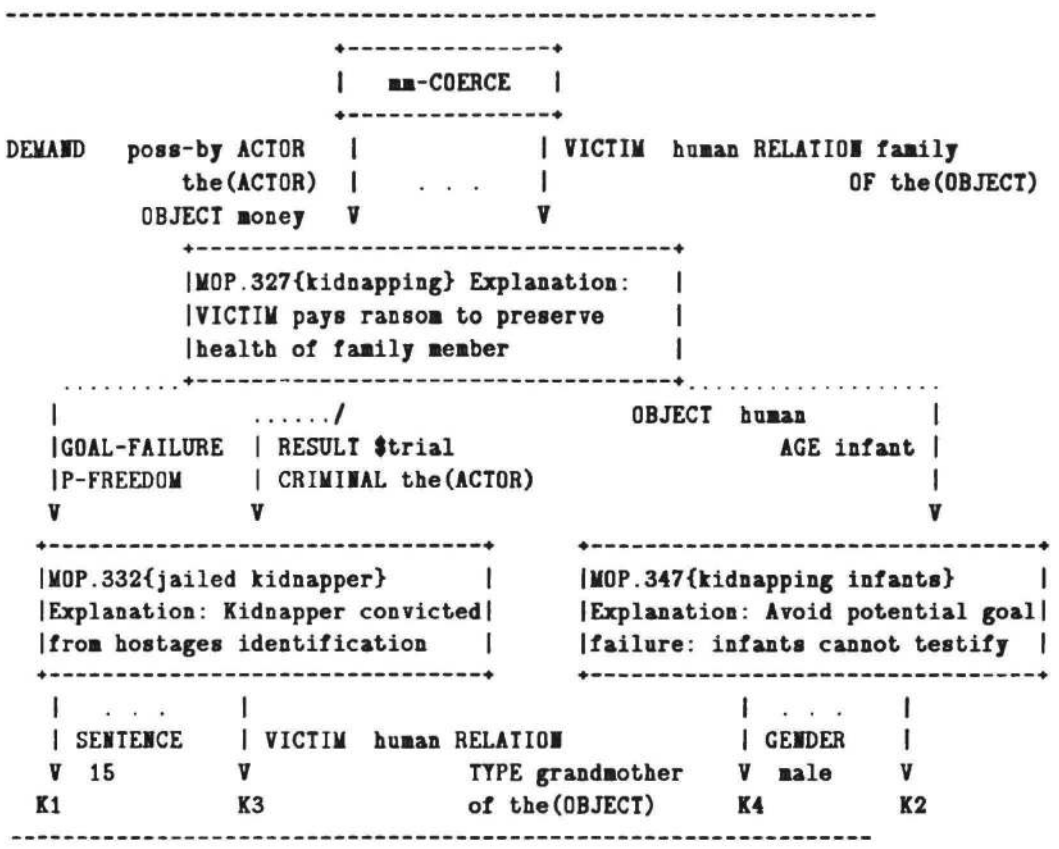

Figure 4: Memory after creating 3 specializations of mm-COERCE

The next event added to memory is K3, which is similar to K1 in that the kidnapper goes to jail after the hostage testifies. There are minor differences in the features of the participants. MOP.327, the kidnapping MOP, is the most specific MOP which is not contradicted by K3. It has an additional RESULT which is similar to a result of K1. A rule which states *If there is a RESULT which thwarts a goal, look for an action before the RESULT which enables the RESULT* finds an inherent flaw in kidnapping: the hostage sees the kidnapper when he is abducted and can testify against the kidnapper. A new MOP, MOP.332 (jailed-kidnapper) is created an indexed under MOP.327 (kidnap) by the indices of the RESULT, the goal failure, and the PREParation which enables the goal RESULT which thwarts the goal as shown in Figure 4. K1 and K3 are indexed under this new MOP, while K2 remains indexed under the kidnapping MOP.

K4, another kidnapping of a blond infant in which the kidnapper was not caught, is added to memory next. MOP.327 (kidnap) is found to be the most specific MOP which describes K4. A similarity is noticed between K4 and K2, the OBJECTs are both blond infants. An applicable generalization rule states *If the PREParation is performed on the an object, look for other MOPs which have a goal failure. Check if the PREParation avoids the goal failure, if it does postulate the ACTOR performed the PREParation to avoid the goal failure.* In this example, the goal of preserving freedom of the kidnapper cannot be thwarted by the infant testifying. A new MOP is created indexed by the AGE of the OBJECT (and not the hair color) as shown in Figure 4.

These examples illustrate the process of creating an explanatory generalized event. In a more realistic set of examples, several organizational generalized events would also be created and each MOP would index a greater number of events and sub-MOPs. The point of creating explanatory generalizations is to create specialized explanations for situations. With only mm-COERCE in memory, the explanation of kidnapping an infant would be "The ACTOR wants the VICTIM to do something" After MOP.327

(kidnap) is created, the explanation would be "The ACTOR wants a member of the OBJECT's family to give him money" After MOP.347 (kidnapping infants) is created the explanation would be "The ACTOR wants a member of the OBJECT's family to give him money and the ACTOR wants to avoid being convicted, so he's kidnapping an infant since infants can't testify"

Conclusion

OCCAM is a program which organizes memories of events and learns by creating explanatory generalized events. It addresses the issue of deciding which features are relevant in producing a generalization. It answers this question by proposing the relevant features are those which are essential in explaining why the event occurred (e.g., why a goal fails). The features which are not essential to arriving at an explanation are exactly those features which would be expected to vary in future events. The unessential features are not used as indices by OCCAM since they are not useful in understanding future events. OCCAM can learn more quickly and accurately than many previous systems since it relies on an explanation process to eliminate unessential features rather than correlation over a large number of examples. Indexing by relevant features has some implications for expert systems which operate by recalling similar experiences. Should a medical expert system index a case by the patient's weight, height, clothing or jewelry? The answer proposed here is to use these as indices in explanatory generalized events only if they are essential in establishing a pathological explanation. Organizational generalized events describe those situations where a coincidence is noted but there is no explanation. These coincidences might initiate and focus the search for new pathological knowledge.

Acknowledgements

Discussions with Mike Dyer and Margot Flowers helped in the evolution of the ideas in this paper. Communications with Michael Lebowitz were also fruitful. This research was supported in part by a grant from the Keck Foundation.

References

1. DeJong, G. Acquiring Schemata Through Understanding and Generalizing Plans. Proceedings of the Eighth International Joint Conference on Artificial Intelligence, Karlsruhe, West Germany, 1983.

2. Dyer, M.. *In Depth Understanding.* MIT Press, 1983.

3. Hayes-Roth, F. and McDermott, J. Knowledge Acquisition from Structural Descriptions. Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, Mass., 1977.

4. Kolodner, J. *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model.* Lawrence Erlbaum Associates, Hillsdale, NJ., 1984.

5. Lebowitz, M. Generalization and Memory in an Integrated Understanding System. Computer Science Research Report 186, Yale University, 1980.

6. Lebowitz, M. "Correcting Erroneous Generalizations". *Cognition and Brain Theory 5*, 4 (1982).

7. Schank, R. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People.* Cambridge University Press, 1982.

8. Vere, S. Induction of Concepts in the Predicate Calculus. Proceedings of the Fourth International Joint Conference on Artificial Intelligence, Tbilisi, USSR, 1975.

9. Wilensky, R. Understanding Goal Based Stories. Computer Science Research Report 140, Yale University, 1978.