Skorstad, J., Gentner, D., & Medin, D. (1988). Abstraction processes during concept learning: A structural view. *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, 419-425.

ABSTRACTION PROCESSES DURING CONCEPT LEARNING: A STRUCTURAL VIEW

Janice Skorstad
Computer Science Department
University of Illinois

Dedre Gentner
Psychology Department
University of Illinois

Doug Medin
Psychology Department
University of Illinois

INTRODUCTION

Learning concepts from examples is a central process in cognition. In the psychological literature, this process is known as *schema-abstraction.* We focus here on the phenomenon of sequence effects, ie., the effect presentation order has upon the concept learned. This paper describes a computer model, SEQL, which provides an experimental tool for exploring a class of schema-abstraction theories. We describe the organization of SEQL, illustrating how structural comparisons combined with a library of abstraction strategies can model sequence learning effects. We briefly describe a psychological experiment designed to explore sequence effects and show how SEQL can be used to reproduce the results.

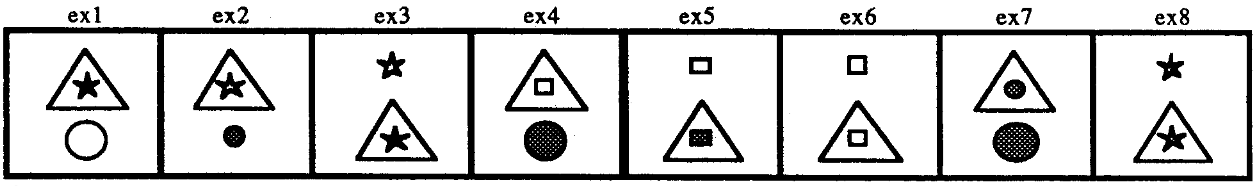
Schema-abstraction theories assume that some information is abstracted during learning and is subsequently stored for later use during classification of new examples. These theories can be differentiated on the basis of three criteria: how the abstracted concept information is (i) **characterized,** (ii) **retained** and (iii) **utilized** to classify new instances. For example, the *prototype* theory [Posner & Keele 68, Posner & Keele 70] assumes that during learning, subjects construct a single representation of the concept's prototype by calculating the average of all the training instances. New exemplars are classified by determining how similar they are to the prototype. Posner & Keele found that classification of never-studied prototypes was more accurate than classification of never-studied exemplars. In fact, after delay there was a greater loss in classification accuracy for the much-studied exemplars than the never-studied prototypes.

In contrast, Medin & Schaffer [Medin & Schaffer 78] proposed an *instance-only* model of concept-learning. This model posits that no abstraction is performed, and only training exemplars are stored. Classification of new training instances is based on their similarity to all of the stored items. Medin & Schaffer demonstrated that the effects arising from schema-abstraction models-namely, the superior performance of prototypes and the increase in this effect after delay, can be accounted for by their instance-only model. In view of this, it has proved difficult to predict differences in the performance of schema-abstraction versus instance-only models.

One phenomenon that might allow us to differentiate these models (and also subclasses of schema-abstraction models) is sequence effects. Such effects are impossible for a standard instance-only model since transfer stimuli are compared to all of the items in memory. These effects therefore might provide evidence for a schema-abstraction model. Unfortunately, it is difficult to predict the performance of schema-abstraction models in concept learning since we lack an explicit definition of the abstraction process.

We aim to attack the problem on two fronts, psychological and computational. We have designed a human experiment and a computer simulation to examine possible abstraction processes.

Group I Sequence:



Group I Expected Rule:

(IF (AND (OUTSIDE X TRIANGLE)
(EQUALS (SHAPE X) CIRCLE))
(EQUALS (LOCATION TRIANGLE) TOP))

(If the figure outside of triangle is a circle, then the triangle is on top.)

Figure 1.

Kline [Kline 83] demonstrated that human learning is indeed affected by presentation sequence. So far, our attempt to replicate Kline has produced weak results. In spite of this, we describe our experiment in order to present the logic behind our reasoning and to show the materials we used.

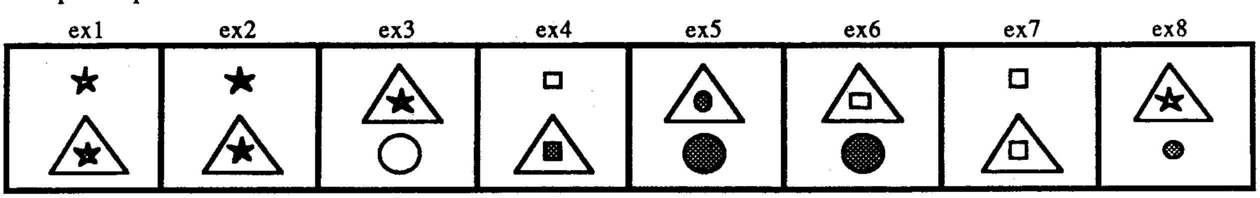
Psychological experiment

This experiment is the first in a series designed to determine if the order of training instances affects the learner's concept description. Subjects first took part in a study phase followed by a transfer phase. Subjects were divided into two groups. We diverged from the standard experimental paradigm since both groups did not simultaneously see the same training instances. Using a method similar to Kline's, each group saw the <u>same</u> training instances but the <u>order</u> of these instances varied between the two groups. Figures 1 & 2 show the two sequences of training instances along with the concept descriptions we expected subjects to generate. Unlike Kline's experiments, subjects did not have full memory; once a new exemplar was displayed, they could not go back and look at previous ones.

Experiment design

Study Phase: Subjects were first told that they would be asked to generate a rule for the figures they were about to see. They were shown 8 figures, one at a time. They were then asked to describe the rule for the figures they just saw. This was repeated four times.

Group II Sequence:



Group II Expected Rule:

(IF (AND (SAME (SHAPE X) (SHAPE Y))
(SAME (SIZE X) (SIZE Y)))
(EQUALS (LOCATION TRIANGLE) BOTTOM))

(If the figures inside and outside of triangle are the same shape and size, then triangle is on bottom.)

Figure 2.

Transfer Phase: Subjects saw 27 pairs of stimuli. For each pair, they were asked to mark which of the two was more similar to the items they saw in the study phase. Subjects saw the same 27 pairs again. (Subjects were told that these were the same 27 pairs they had just seen.) This time, in addition to marking which was more similar to study items, subjects were asked to explain their choice.

Results

The results were not strong, nor were they uniform within the groups. The difference between the groups was not significant. However, group 2 performed as expected; 55% of group 2 subjects chose rule 2. Group 1 subjects showed no preference for rule 1 over rule 2. An ANOVA did reveal a significant difference between the two rules (**). Rule 2 was found to be easier to learn than rule 1. In light of these results, which are somewhat promising, we modified the training instances so that rule 1 would be as easily learned as rule 2. [Skorstad 88] will provide a more detailed description of the experiment and its follow-up.

SEQL computer model

SEQL is being built as a tool for exploring various abstraction theories. Its results can be compared with human performance in the learning experiment. Our approach differs from many prior schema-abstraction theories. We assume that the prototype must be based on structural similarity comparisons rather than feature-set intersection. We use a model of similarity as defined by structure-mapping theory [Gentner 83]. For this purpose we use SME [Falkenhainer, Forbus & Gentner 86], a computer implementation of the structure-mapping theory.

System Description

SEQL consists of several main modules:

— **SME** operates on two potential analogs, the *base* and the *target,* generating a number of plausible mappings *(gmaps)* for these analogs. These gmaps correspond to interpretations of the analogy. In addition, SME produces an evaluation score for each of the gmaps. An important feature of SME is its toolkit approach. By using different sets of match rules, we can perform the various similarity comparisons defined by structure-mapping theory. In this research, we use the Literal-Similarity rules. By literal-similarity, we mean a similarity match in which both the objects and the relational structure of these objects are counted in the match.

— **SE** (Structural Evaluator) is a post-SME process. It provides alternate gmap evaluation scores, separate from SME's evaluator. For example, one can choose to evaluate a gmap based on a combination of its depth and breadth or on its depth and breadth <u>relative</u> to the depth and breadth of its base.

— **Generalize** finds the most specific conjunctive generalization that characterizes both the base and target descriptions fed to SME. Generalize takes a single gmap as input and outputs a generalized description of this gmap both in human readable form and in SME's syntax.

— **Specialize** modifies a generalization when it is no longer adequate to describe a new example. This occurs when the generalization is poorly matched to a target example. If the generalization's rule description (the human readable form) is conjunctive, then the generalization must be specialized. It needs to be modified in such a way so that it covers a smaller subset of examples. This can be done by transforming the conjunctive rule description into an IF-THEN rule roughly in the style proposed by Bettger [Bettger, in preparation]. If the generalization rule was already in IF-THEN form, no specialization is necessary.

Library of abstraction strategies to be simulated

In this section, we lay out a space of possible abstraction strategies. This library of strategies will give us the flexibility to explore various abstraction theories. These strategies determine what is generalized, and when this generalization takes place. We divide the abstraction strategies into two classes, the **limiting cases** versus the **combination models.** First we list a set of limiting cases which are useful in delimiting the space of basic abstraction theories. These models are probably too simple to be used for direct comparison to human performance during the study phase. Their effectiveness or power becomes more apparent during the transfer task when the stored knowledge is retrieved and utilized to classify new instances. Next, we list the combination models. These use a parameter, T, the threshold value. If the evaluation score of a match is below T, the match is not considered to be a good one. In this way, the user has control over what he considers to be a good match.

A. Limiting Case Models
1. Exemplar Only
   Store exemplars only. No generalizations are made during learning. This is the simplest form of the instance-only model.
2. Radical Generalization
   Continuously generalize each new exemplar with the base (which is a generalization) to generate one single, "winning" generalization. Some versions of the prototype models would fall under this category.
3. All Exemplars & All Generalizations
   Store all exemplars and all generalizations (which are formed for each match). This puts a heavy burden on memory. If there are n items in the study sequence, (2n - 1) items will be stored.

B. Combination Models
1. Generalization and Exemplar
   If the match is greater than T, store their generalization, the base, and the target, otherwise store the base and target only (up to a memory limit).
2. Generalization or Exemplar
   If the match between the base and target is greater than T, store their generalization, otherwise store the base and target (up to a memory limit).
3. IF-THEN-Rule-Generator
   This is a specialization of model B.2. In addition to storing the base and target when a bad match is found, the generalization rule is modified to form an IF-THEN rule. The psychological motivation for this strategy is the intuitive notion that when humans encounter an exception during concept learning, ie., when an example is found which isn't covered by their rule, they focus on the differences in the base and target. They use these features to patch up their current rule description [Medin personal communication 87 & Bettger in preparation].

One of the goals of our research is to test each of these abstraction strategies on the same set of examples. We illustrate the **IF-THEN-Rule-Generator** strategy in more detail in the next section.

Ex1	Ex2	Gen[1,2]
(star obj1)	(star obj1)	(star entity17)
(star obj2)	(triangle obj2)	(star entity24)
(triangle obj3)	(star obj3)	(triangle entity33)
(small obj1)	(small obj1)	(small entity17)
(small obj2)	(large obj2)	(small entity24)
(large obj3)	(small obj3)	(large entity33)
(above obj1 obj2)	(above obj1 obj2)	(above entity17 entity24)
(above obj1 obj3)	(above obj1 obj3)	(above entity17 entity33)
(inside obj2 obj3)	(inside obj3 obj2)	(inside entity24 entity33)
(outside obj1 obj3)	(outside obj1 obj2)	(outside entity17 entity33)
(bottom obj3)	(bottom obj2)	(bottom entity33)
(same-shape obj1 obj2)	(same-shape obj1 obj3)	(same-shape entity17 entity24)
(same-size obj1 obj2)	(same-size obj1 obj3)	(same-size entity17 entity24)

Figure 3.

SEQL example using IF-THEN-Rule-Generator strategy

Suppose our goal is to generate a concept description for the sequence of 8 exemplars shown in figure 2. The representations for exemplar 1 and 2 are shown in figure 3. (For compactness, we removed some of the attributes). Ex1 and ex2 are input to **SME. Ex1** is the base and ex2 the target. SME generates a number of plausible mappings (gmaps) between these two exemplars. Running the Structural Evaluator on these mappings yields a single best normalized gmap which is compared with the threshold value to see if it is a good match. For these two exemplars, we have a good match. The gmap is passed on to the generalizer which produces the generalization, gen[1,2] shown in figure 3.

The generalization, gen[1,2] and the new exemplar, ex3, (see fig 4) become the next base and target respectively. They don't yield a good match. The highest scoring normalized gmap is less than the threshold value. **Gen[1,2]'s** rule description must be specialized since it no longer describes all of

Gen[1,2]	Ex3	IF-THEN Rule (gen [1,2], ex3)
(star entity17)	(star obj1)	(IF (AND (star entity24)
(star entity24)	(circle obj2)	(small entity24)
(triangle entity33)	(triangle obj3)	(above entity17 entity24)
(small entity17)	(small obj1)	(inside entity24 entity33)
(small entity24)	(medium obj2)	(same-shape entity17 entity24)
(large entity33)	(large obj3)	(same-size entity17 entity24))
(above entity17 entity24)	(bottom obj3)	(THEN
(above entity17 entity33)	(above obj1 obj2)	(AND (triangle entity33)
(inside entity24 entity33)	(above obj3 obj2)	(large entity33)
(outside entity17 entity33)	(inside obj1 obj3)	(star entity17)
(bottom entity33)	(outside obj2 obj3)	(small entity17)
(same-shape entity17 entity24)		(above entity17 entity33)
(same-size entity17 entity24)		(outside entity17 entity33)
		(bottom entity33))))

Figure 4.

```
(IF  (AND  (small entity24)
           (inside entity24 entity33)
           (same-shape entity17 entity24)
           (same-size entity17 entity24))
  (THEN
   (AND    (triangle entity33)
           (large entity33)
           (above entity17 entity24)
           (small entity17)
           (above entity17 entity33)
           (outside entity17 entity33)
           (bottom entity33))))
```

IF-THEN Rule (sequence 2)

```
(IF  (AND  (circle entity64)
           (above entity12 entity64)
           (above entity70 entity64)
           (outside entity64 entity12))
  (THEN
   (AND    (triangle entity12)
           (large entity12)
           (top entity12)
           (small entity70)
           (inside entity70 entity12)))
```

IF-THEN Rule (sequence 1)

Figure 5.

the exemplars seen. Specializing gen[1,2] produces the new rule description, **IF-THEN Rule (gen[1,2], ex3)** shown in figure 4. This description asserts that if there is a small star inside another object, with an object of the same shape and size above it, then there is large triangle located at the bottom of the frame, with a small star above and outside it. This human-readable IF-THEN rule is kept in addition to the original gen[1,2] assertions. In the next match, it is these assertions, gen[1,2], that are input to SME-not the IF-THEN rule. Running the remainder of the exemplars through SEQL and displaying the generalization with the highest weight, we get the concept description, **IF-THEN Rule (sequence** 2). (see figure 5)

IF-THEN Rule (sequence 2) asserts that if there is a small object inside another object, with an object of the same shape and size above it, then there is large triangle located at the bottom portion of the frame, with a small object above and outside it. This description corresponds to the description that 55% of the subjects from group 2 of our experiment generated.

When the <u>same</u> set of exemplars are presented to SEQL in a <u>different</u> order, a different rule description is generated, thus demonstrating a sequence effect. For example, if the sequence shown in figure 1 is run through SEQL, the rule **IF-THEN Rule (sequence 1)** listed in figure 5 is . produced. This rule states that if there is a circle which is below two entities and outside of one of the entities, then there is a large triangle located at the top of the frame with a small object inside it.

SEQL's results are considerably "better" than those generated by our human subjects, especially those in group 1. Hopefully, the modifications we're making to our experiment will improve subjects' performance. If not, this will discount the psychological validity of our model.

Discussion

Current work in concept-formation suggests that abstraction does indeed take place during concept learning [Ross et. al. submitted for publication 87, Elio & Anderson 84, Kline 83]. Our delineation of a space of some of these possible abstraction processes enables us to capture and avoid some of the extremes of the prototype and instance-only models.

An important distinction of our approach is the use of. structural comparisons defined by structure-mapping theory as opposed to the more typical feature-set intersection. We have shown how SEQL incorporates structural comparisons to produce a useful tool for exploring abstraction processes that occur during concept learning.

Finally, we have shown how SEQL can simulate the phenomenon of sequence effects.

Acknowledgements

Ken Forbus & Gordon Skorstad provided valuable suggestions, both in the research and in the writing of this paper. Brian Falkenhainer provided many useful comments and suggestions on the research. This research is supported by the Office of Naval Research, Contract No. N00014-85-K-0559.

References

Bettger, J. (in preparation), Master's Thesis, University of Illinois, Department of Psychology.

Elio, R., & J.R. Anderson (1981), "The Effects of Category Generalizations and Instance Similarity on Schema Abstraction", *Journal of Experimental Psychology: Human Learning and Memory*, **7** (6), 397-417.

Elio, R., & J.R. Anderson (1984), "The Effects of Information Order and Learning Mode on Schema Abstraction", *Memory & Cognition*, **12** (1), 20-30.

Falkenhainer, B., K.D. Forbus, D. Gentner (1986), "The Structure-Mapping Engine", *Proceedings of the Fifth National Conference on Artificial Intelligence*, August.

Gentner, D. (1983), "Structure-Mapping: A Theoretical Framework for Analogy", *Cognitive Science* 7(2).

Hayes-Roth, F., & J. McDermott (1978), "An Interference Matching Technique for Inducing Abstractions", *Communications of the ACM*, **21**, 5, 401-411

Kline, P.J. (1983), "Computing the Similarity of Structured Objects by Means of a Heuristic Search for Correspondences", Doctoral Dissertation, University of Michigan, Department of Psychology.

Medin, D.L., & M.M. Schaffer (1978), "Context Theory of Classification Learning", *Psychological Review*, **85** (3), 207-238.

Posner, M.I., & Keele, S.W. (1968), "On the genesis of abstract ideas", *Journal of Experimental Psychology*, **77**, 353-863.

Posner, M.I., & Keele, S.W. (1970), "Retention of Abstract Ideas", *Journal of Experimental Psychology*, **83**, 304-308.

Ross, B.H., S.J. Perkins & P.L. Tenpenny, (submitted for publication, 1987), "Reminding-Based Category Learning".

Skorstad, J.C. (in preparation), Master's Thesis, to appear as a Technical Report, University of Illinois, Department of Computer Science.