On the Indistinguishability of Exemplar Memory and Abstraction in Category Representation

Lawrence W. Barsalou
Georgia Institute of Technology

In the target article for this volume, Eliot Smith contrasts two kinds of category representation: exemplar memories and abstractions. After reviewing a wide variety of studies, Smith concludes that exemplar memories play a central role in representing social categories. However, numerous complexities surround claims about representation, and identifying the representation that underlies performance on a particular task is a formidable challenge. In this article, I explore the characteristics of exemplar and abstracted representations and assess our ability to distinguish them empirically.

In the first section, I briefly review work on category learning. To some extent, misunderstandings about category representation reflect a failure to appreciate this history. In the second section, I discuss two issues essential to evaluating representation: (1) the necessity of considering processing; (2) the distinction between information and storage. I then present three dimensions of information storage that are useful in formally assessing representation: (1) information duplication, (2) information revision, and (3) information loss. In the third section, I present three fallacies about abstraction. These fallacies are not only present in Smith's article but seem to predominate how readers interpret work on category learning. In the fourth section, I show how exemplar and abstracted representations in principle are informationally equivalent. Because of this equivalence, we can not determine whether people use exemplar or abstracted representations. In the fifth section, I present four general classes of category learning models: permanent trace models, revisable trace models, cumulative abstraction models, and reductive abstraction models, the last of which includes connectionist models. As is shown, once one considers the wide variety of possible representations in these models, it becomes increasingly difficult to
2. EXEMPLAR MEMORY AND ABSTRACTION

Roth & Hayes-Roth, 1977; Neumann, 1974; Reitman & Bower, 1973). Instead the primary goal of exemplar theorists has been to show that certain exemplar models are at least as effective as the early abstraction models that dominated thinking in the area—it has not been to reject the entire class of abstraction models.

Readers of the category learning literature often perceive otherwise. In his target article, Eliot Smith concludes, “Research on nonsocial categorization thus supports the generalization that specific category exemplars are stored and used in making category membership judgments” (p. 11). The distinction between exemplar and abstracted representations is sufficiently salient and seductive that it has misguided interpretation of work in the area.

REPRESENTING EXEMPLARS AND ABSTRACTIONS

To assess the role of representation in learning, it is necessary to consider the relation between representation and processing, the distinction between information and storage, and the dimensions that define exemplar and abstracted representations.

Representation and Processing

Cognitive theorists widely agree that any model of human performance must contain assumptions about both representation and processing. Because we cannot observe people’s representations of knowledge directly, we cannot test claims about representation in the absence of processing mechanisms. Instead we can only observe the effects of a representation as they occur through processes that operate on it. Whatever behaviors we observe to assess a representation necessarily reflect processing as well.

It is therefore impossible to conclude from behavioral research on category learning that people represent categories with exemplars or abstractions. Instead we can only conclude that particular models (i.e., representation-process pairs) are either supported or rejected. For example, data may support a model that assumes a particular exemplar representation and particular processes that operate on it. Other models with exemplar representations may be rejected because their combination of representation and processing assumptions provides incorrect predictions. Analogously, abstraction models may succeed or fail, depending on the joint adequacy of their representation and processing assumptions. Similar arguments about representation and processing have been made by Anderson (1978) and Palmer (1978).

---

1 Although I focus on Smith’s arguments about exemplar and abstracted representations, his arguments about declarative and procedural representations suffer a similar predicament. Conclusively determining that certain findings support declarative learning and that others support procedural learning is difficult if not impossible (Winograd, 1975).

2 This brief review barely begins to cover the large literature on category learning. For more extensive reviews, see Smith and Medin (1981), Mervis and Rosch (1981), Medin and Smith (1984), and Oden (1987).
Information Versus Storage

Although we cannot empirically assess the psychological presence of exemplar or abstracted representations in the absence of processing, we can address their formal and computational properties. For example, we can try to distinguish exemplar and abstracted representations in terms of the information they contain. We might suppose that abstracted representations contain only properties that occur definifyingly or characteristically across category members; whereas exemplar representations also contain idiosyncratic information that individuates exemplars. Consider Fig. 2.1. Imagine that a person experiences five exemplars from a category (e_1 through e_5), each having three properties (e.g., e_1 has properties a, b, and d). These properties can represent contextual information and operations performed on an exemplar, as well as its physical characteristics. Although only three properties are shown for each exemplar, exemplars may certainly contain more.

As a result of experiencing e_1 through e_5, a person might develop an abstracted representation for the category, such as A_1 in Fig. 2.1. As can be seen, this abstracted representation only contains characteristic properties, that is, properties generally true of exemplars (i.e., a, b, and c), along with their frequency of occurrence (i.e., 4, 3, and 3). Idiosyncratic properties that individuate exemplars have been discarded (e.g., d for e_5). In contrast, consider exemplar representation E_1. As can be seen, this exemplar representation contains idiosyncratic properties of exemplars (e.g., d for e_5). A_1 and E_1 represent standard assumptions about the information in abstracted and exemplar representations.

But consider representations A_2 and E_2. A_2 is an abstracted representation that maintains idiosyncratic information about exemplars (e.g., d from e_1). E_2 is an exemplar representation that loses idiosyncratic information about exemplars, after storing a complete record of the first two. From the third exemplar on, information is only stored in an exemplar memory if it occurs in a previous exemplar. Although the point at which this transition occurs is arbitrary here, it could be well-motivated in an actual model. Consequently, one cannot distinguish between exemplar and abstracted representations in terms of the information they contain. Either type can contain or discard idiosyncratic information.

If information does not distinguish exemplar and abstracted representations, then what does? The distinguishing factor is information storage. In the next section, I present three dimensions that structure how information can be stored as exemplars are experienced. These dimensions serve as the basis for how I define exemplar and abstracted representations.

Dimensions of Information Storage

Three dimensions reflect important differences in how information is stored in exemplar and abstracted representations. They are information duplication, information revision, and information loss. By no means are these the only dimensions that describe representations.

*Information Duplication.* Perhaps the primary distinction between exemplar and abstracted representations concerns information duplication. Exemplar representations exhibit information duplication, whereas abstracted representations do not. As stated by Eliot Smith, “classifying a single category instance leaves a trace in memory” such that “our knowledge of a category . . . is (at least in part) distributed in memory across representations of a series of specific instances, rather than embodied in a single prototype representation” (p. 10).
Consider Fig. 2.1 again. As can be seen from exemplar representation $E_1$, property information is duplicated across exemplar memories. For example, property $a$ occurs four times, once for each of the representation of $e_1$, $e_2$, $e_4$, and $e_5$. Every time a new exemplar containing $a$ is stored, another duplication of $a$ occurs. This allows exemplar information to remain independent. Each exemplar memory contains completely separate information, with information for a given property being distributed across exemplar memories.

In contrast, abstracted representations centralize property information. Each property is only represented once, even if it occurs across many exemplars. As can be seen from representation $A_1$, $a$ is only represented once, even though it occurs in four exemplars. Every time a given property is encoded, the same memory structure is processed. Abstraction models integrate exemplar information by updating centralized property information.\(^3\)

*Information Revision.* Another key difference between exemplar and abstracted representations concerns information revision. In ideal exemplar models, as discussed shortly, an exemplar representation is never revised. Every processing episode produces a new exemplar memory that remains permanently in the system. Previous exemplars may be retrieved, and information from them may control decision making and become part of a new exemplar memory. However, the retrieved exemplar memory must remain unchanged, else information is lost from memory. In general, ideal exemplar models assume that exemplar memories are not revised by subsequent processing (e.g., Bekner & Bowers, 1983; McCloskey & Zaragoza, 1985).

An exemplar model can certainly allow revision of exemplar memories, and as described later, some do. But such a model may begin to lose its exemplar character and become more like an abstraction model. Consider an exemplar memory that is retrieved on many occasions to help process new members of its category. Across occasions, properties in the memory could be revised, with properties frequently relevant to categorizations being added, and with properties never relevant to categorizations being deleted (e.g., Loftus, 1975). Enough such processing could eventually transform the exemplar memory into a representation of the category's central tendency. Although the exemplar representation once contained idiosyncratic information, it now contains only characteristic information.

Whereas information revision is optional for exemplar representations (depending on the accompanying processes), it is intrinsic to abstracted representations. Consider representation $A_2$ in Fig. 2.1. Every time a new exemplar is encoded, all the relevant properties must be revised. If $acf$ is encoded, for example, the properties for $a$, $c$, and $f$ are incremented by 1. The essential operation of an abstraction model is to revise centralized category information.

*Information Loss.* Typically, theorists assume that exemplar models lose less information than abstraction models, as represented by the contrast between $E_1$ and $A_1$ in Fig. 2.1. As we see later, however, this factor does not really distinguish exemplar from abstracted representations. Each type of representation may not lose any information at all, instead containing a complete record of all exemplar information. Or each type of representation may lose substantial amounts of information. Nevertheless, the dimension of information loss is central to distinguishing different representations, as we shall see.

Information may be lost from a representation either through intentional revision or incidental degeneration. In intentional revision, a processor intentionally alters the contents of an exemplar memory or abstraction. In accidental degeneration, the contents of an exemplar memory or abstraction are partially or totally lost due to side effects of other processes (e.g., interference, decay).

In summary, my definitions of *exemplar representation* and *abstracted representation* from here on reflect the following assumptions about information storage: Exemplar representations exhibit information duplication and may exhibit no information revision, or at least much less revision than occurs for abstracted representations. In contrast, abstracted representations centralize property information and require information revision, because every new exemplar causes centralized property representations to be updated. Information loss does not distinguish these two representations. As we shall see, either may lose no information or may lose information to equivalent extents.

It is important to note that some theorists define exemplar and abstraction models in terms of processing, rather than representation. These theorists adopt an *identical* exemplar representation for *all* models, but then define individual models as "exemplar" or "abstraction" in terms of the decision processes that operate on the exemplar representation (e.g., Estes, 1986; Koh, 1989; Koh & Meyer, 1989). For example, these abstraction models compare the exemplar being classified to an abstraction constructed from existing exemplar representations during classification. In contrast, these exemplar models compare the exemplar being classified to the same exemplar representations individually. Defining "exemplar" and "abstraction" in terms of processing is certainly useful. But because both kinds of processing can operate on either exemplar or abstracted representations, as I define them, this approach is somewhat orthogonal to the representational issue I pursue here. It would be useful if future treatments of the exemplar-abstraction issue assessed its application to representation and processing more systematically.
FALLACIES ABOUT ABSTRACTION

Given these basic assumptions of exemplar and abstracted representations, we can begin to assess claims about them. In this section, I consider three fallacies that reflect false stereotypes about abstraction.

Fallacy 1: Abstractions Do Not Contain Idiosyncratic Information

It is often believed that the knowledge abstracted for a category only contains properties that are defining or generally true of category members. Eliot Smith describes abstractions as representing the typical characteristics of a class of objects or events, rather than the details of a specific experience. According to this view, idiosyncratic properties that distinguish exemplars are discarded at encoding and do not exist in the category representation.

But as we saw earlier in Fig. 2.1, an abstracted representation can contain idiosyncratic information about exemplars (e.g., representation $A_2$). There is no a priori reason why idiosyncratic information cannot be abstracted from exemplars and integrated into a centralized category representation. An abstracted representation can contain any property occurring in any exemplar, regardless of how often it occurs. Such information might be useful in making later categorizations, assuming that certain idiosyncratic properties occur for one category and not other. The less information discarded, the more optimal categorization is likely to be, generally speaking. No a priori restriction limits the properties in a centralized category representation to those that are defining or generally true.

Reitman and Bower (1973), Neumann (1974), and Hayes-Roth and Hayes-Roth (1977) have all proposed abstraction models that store idiosyncratic information about exemplars in this manner. As discussed later, connectionist models also store idiosyncratic information in the process of abstracting information from exemplars (e.g., McClelland & Rumelhart, 1985).

Fallacy 2: Abstractions Do Not Contain Coocurrence Information

It is often believed that the knowledge abstracted for a category does not contain information about correlations between properties. Eliot Smith states that "correlated attributes play no special role" when abstracted knowledge is used in categorization decisions. Instead only information about each property’s relative frequency of occurrence is stored, independent of how often it cooccurred with other properties. According to this view, if someone experiences exemplars $e_6$ through $e_{10}$ in Fig. 2.2, only the independent frequencies of properties are abstracted, as exhibited by $A_3$. Information about how often properties cooccur has been discarded.

However, there is no a priori reason why cooccurrence information cannot be abstracted as well. As represented by $A_4$ in Fig. 2.2, the knowledge abstracted to represent $e_6$ through $e_{10}$ could contain all the cooccurrence information available across exemplars. Information about pairwise cooccurrence is represented by the integers on the left of $A_4$. For example, the 2-tuple $ab$ occurred in 2 exemplars, and the 2-tuple $cf$ occurred in 1 exemplar. Information about property triples is represented on the right of $A_4$. For example, the 3-tuple $abd$ occurred in 2 exemplars, and the 3-tuple $bcf$ occurred in 1 exemplar. Consequently, abstracted
representations can record cooccurrence information by storing the frequency of higher-order \( n \)-tuples. \( A_n \) also represents independent property frequencies (1-tuples), as shown by the integers below the properties.

If cooccurrence information is diagnostic for categorization, an abstraction model may utilize this information in categorization decisions. Again, the less information discarded, the more optimal categorization is likely to be, generally speaking. The models of Reitman and Bower (1973), Neumann (1974), Hayes-Roth and Hayes-Roth (1977), and Elio and Anderson (1981) all abstract cooccurrence information from exemplars, as do connectionist models (e.g., McClelland & Rumelhart, 1985).

Abstraction models that store cooccurrence information have sometimes been criticized by exemplar theorists as making unreasonable storage demands (Medin & Schaffer, 1978). As can be seen from \( A_n \) in Fig. 2.2, the number of \( n \)-tuples used to represent all possible cooccurrences can quickly become large (i.e., \( 2^n - 1 \)). For example, if an exemplar contains 20 properties, it takes 1,048,575 \( n \)-tuples to represent it. However, the following six points must be considered in evaluating this problem. First, the number of properties encoded for an exemplar may typically be small, such that a reasonable number of \( n \)-tuples is required to store it. For example, if the limited capacity of working memory affects exemplar encoding, then five properties may typically be acquired for each exemplar, thereby requiring 32 \( n \)-tuples (cf. Miller, 1956). Second, no one has yet noted a limit on human memory, and the human cognitive system may well have the capacity to store substantial amounts of cooccurrence information. Third, it is essential to compare the relative number of exemplars to properties before concluding that exemplar models are more storage efficient. For example, imagine that a person experiences 1000 exemplars for a category, all of which have 3 properties drawn from \( a, b, c, d, e, \) and \( f \) (as for \( e_6 \) through \( e_{10} \) in Fig. 2.2). In this case, the total number of representational units, roughly speaking, is 1000 for an exemplar model (1 unit per exemplar) but only 41 for an abstraction model (6 units for all possible 1-tuples, 15 units for all possible 2-tuples, 20 units for all possible 3-tuples). In some cases, abstraction models that record cooccurrence information require less storage capacity than exemplar models (but not in all cases). Moreover, exemplar models are certainly not storage efficient, relatively speaking, because they often lose little if any information about exemplars. Fourth, it is very unlikely that all possible combinations of properties occur across exemplars, especially if properties correlate highly. In \( A_n \), for example, 2-tuples do not exist for \( ab \) and \( bc \), and 3-tuples do not exist for \( abc \) and \( bce \). If properties form attribute-value structures, this can further reduce the set of possible correlations, because values for the same attribute are often mutually exclusive (i.e., they never cooccur). Even if all perceived cooccurrences are stored, not all possible cooccurrences must be represented. Fifth, abstraction models that extract only a subset of possible \( n \)-tuples may often compete favorably with exemplar models, as we shall see at several points later. Sixth, and perhaps most importantly, people may only store combinations of properties that receive local attention (Trabasso & Bower, 1968), that enter into systematic patterns of correlation (Billman & Heit, 1988), or that are relevant to intuitive theories (Murphy & Medin, 1985). Such mechanisms greatly reduce the \( n \)-tuples encoded for exemplars.

**Fallacy 3: Abstractions are Static and Unchanging**

It is often believed that knowledge abstracted for a category is static and relatively unaffected by exemplars. Eliot Smith states that "the prototype should only change slowly with exposure to new category instances." Although some abstraction models exhibit this characteristic (e.g., Posner & Keele, 1968; Reitman & Bower, 1973), others do not. For example, abstraction models whose learning rules optimize cue predictability can produce large shifts in stable categorization after an unusual event (e.g., Gluck & Bower, 1988; Rescorla & Wagner, 1972). Moreover, some exemplar models become increasingly entrenched after much learning, namely, those that use all previous exemplars in every categorization decision (e.g., Medin & Schaffer, 1978). Each subsequent exemplar has a smaller and smaller impact on categorization, given the increasing number of other exemplars that influence categorization. Analogous to abstraction models, some exemplar models can produce major performance shifts late in learning, namely, those that only use subsets of exemplar memories (e.g., Reed, 1972), or those whose exemplar memories can be activated to different extents (e.g., Hintzman, 1986).

Abstraction models are often construed as representing categories in a static manner. Every time a category is represented, the same abstraction represents the category, regardless of context. But exemplar models sometimes work this way as well, with all exemplar memories affecting every categorization decision (e.g., Estes, 1986; Medin & Schaffer, 1978; Nosofsky, 1988). If one assumes that the entire exemplar set represents the category, then the category representation does not change from trial to trial, except for the addition of new exemplars. In effect, this is no different than updating an abstraction after each new exemplar.

Other exemplar theorists assume that the target stimulus currently being categorized does not retrieve the entire exemplar set. Instead only those exemplars similar to the target are retrieved to control its categorization, with the target being assigned to the category from which the most exemplars were retrieved. Because the exemplars retrieved for a given category can vary widely across categorizations, representation of the category varies. Assume that \( e_{10} \) in Fig. 2.2 represent the exemplars stored in memory for a category. During the categorization of a target stimulus, imagine that exemplars are retrieved if they share at least one property with the target. Figure 2.2 shows the exemplars that would be retrieved for the targets \( hfg \) and \( ceh \). As can be seen, the exemplars retrieved differ, demonstrating that the category is being represented dynamically during these different categorizations.
However, retrieval from an abstracted representation can also vary in this manner during categorization. Assume that a category is represented by $A_4$ in Fig. 2.2. During the categorization of a target stimulus, imagine that $n$-tuples are retrieved if they share at least one property with the target. Figure 2.2 shows the $n$-tuples that would be retrieved for the targets bfg and ceh. As can be seen, the $n$-tuples retrieved differ, demonstrating that the category is being represented dynamically during these different categorizations. Barsalou's (1987, 1989) theory of category representation works in this manner. Connectionist models also exhibit this dynamic quality (McClelland & Rumelhart, 1985).

**INFORMATIONAL EQUIVALENCE AND ITS IMPLICATIONS FOR DISTINGUISHABILITY**

We have seen that some abstracted representations can support processing often believed to occur only for exemplar representations. But are these two classes of representation completely equivalent? The key issue is assessing their informational equivalence, as I describe next for ideal exemplar and abstraction models. By **ideal**, I do not mean that these models are typical, nor that they are ideal for all purposes. I simply mean that they are ideal in not exhibiting information loss.

**Ideal Exemplar Models**

Consider an idealized case in which all exemplar representations remain accessible in memory. Under these conditions, every exemplar ever experienced is potentially available for use in processing its category. For example, representation $E_3$ in Fig. 2.3 maintains all information from the original exemplars, $e_6$ through $e_{10}$, in Fig. 2.2. No information is lost.

Because no information is lost, a model with $E_3$ as its representation can be made equivalent to any abstraction model. Any abstraction model starts with all the information available in exemplars and then integrates it into a category representation. For example, an abstraction model might construct representation $A_3$ in Fig. 2.2 from $e_6$ through $e_{10}$. However, a model with representation $E_3$ can mimic a model with $A_3$, if it contains a mechanism that can abstract the independent frequencies of properties at retrieval. Similarly, a model with $E_3$ can mimic a model with $A_3$, if it contains a mechanism that can abstract cooccurrence information at retrieval. Any result that an abstraction model explains, an ideal exemplar model can also explain, if it can perform the requisite processes at retrieval.

One might argue that such exemplar models are simply abstraction models that abstract at retrieval instead of encoding. Two points should be noted in this regard. First, it is important to distinguish between models that lose information from models that do not. Exemplar models that abstract at retrieval may never lose information, whereas some abstraction models, such as those with $A_3$, do. If it can be shown that people do not lose information and are able to abstract, then exemplar models that exhibit both properties should be preferred over abstraction models that do not. Second, such exemplar models should ultimately encode abstractions into memory. Exemplar theorists rarely assume that abstractions constructed at retrieval subsequently become stored in memory. But abstractions should be stored, if they receive sufficient processing. This follows from everything we know about the roles of effort, rehearsal, and depth-of-processing in transferring information from working memory to long-term memory. Because abstractions probably require substantial processing resources to construct and use, their entry into long-term memory may occur often. As a result, exemplar models that abstract at retrieval ultimately encode abstractions into memory, much like abstraction models.

**Ideal Abstraction Models**

It is easy to construct ideal abstraction models that lose no information about exemplars. In fact, the Reitman and Bower (1973) model, along with certain models in Hayes-Roth and Hayes-Roth (1977) and in Gluck and Bower (1988), are ideal in this sense. Consider $A_4$ in Fig. 2.3, which represents exemplars $e_6$ through $e_{10}$ in Fig. 2.2. As can be seen in Fig. 2.3, all the original exemplar information that produced $A_4$ can be abstracted by decomposing its 3-tuples into the E's shown in $A_4'$. Once this information is available, an ideal abstraction model can produce any operation possible for an exemplar model, because all the original exemplar information is available.

Decomposition, as shown for $A_4$ and $A_4'$, works optimally if all exemplars contain the same number of properties. If exemplars contain different numbers of properties, it becomes complicated, using the representation in $A_4$, to reproduce the original exemplars. Imagine that 2-tuple $ax$ has a frequency of 24 across some set of exemplars. This frequency could have resulted from 6 two-property exemplars that only contain $ax$ and from 18 three-property exemplars that contain $ax$ as a 2-tuple (e.g., $aba$, $axc$). Alternatively, $ax$'s frequency of 24 could have resulted from 18 $ax$ exemplars and 6 three-property exemplars that contain $ax$. To reproduce the original exemplars, an abstraction model could record the frequency of $ax$ as an exemplar, as well as the total frequency of $ax$. For example, if $ax$ occurs 6 times as an exemplar and 18 times as a 2-tuple in three-property exemplars, then $ax-24,6$ enables recovery of the original exemplars. This abstracted representation, and others of its form, exhibit no information loss and are informationally equivalent to ideal exemplar models.

It may not be necessary for abstraction models to store the highest-order $n$-tuples to compete favorably with exemplar models. Nearly every abstraction model that has been rejected by exemplar theorists only stores information about 1-tuples (i.e., prototype models). No information is stored about any level of
because they contain the same information. But as we have seen in Figs. 2.1 and 2.2, information content does not distinguish exemplar and abstracted representations. Instead what distinguishes them is information storage. As defined earlier, exemplar memories are distributed data structures that typically receive little if any revision once stored; whereas abstractions are centralized data structures that receive constant revision. Consequently, the 3-tuples in \( A_3 \) have different theoretical properties than the exemplar memories in \( E_3 \). Although these two representations contain the same information, they store it in fundamentally different manners.

What about empirical distinguishability? Can we determine from behavioral data whether people store category information in exemplar memories or abstractions? First, consider categorization decisions. If we ignore how long a model takes to produce a particular decision, we can always find one exemplar model and one abstraction model that predict the same decisions. Because both models potentially have the same information at their disposal, they can produce identical responses, given appropriate processing assumptions. If data reject a particular abstraction model and support a particular exemplar model, we can always adopt an ideal abstraction model with the processes of the exemplar model to account for the data. We can make a similar move to handle a rejected exemplar model. On the basis of decision data, we can never rule out the entire class of models that uses exemplar representations, nor can we ever rule out the entire class of models that uses abstracted representations.

Can reaction time data distinguish whether people use exemplar or abstracted representations? It is not unreasonable to believe so. For example, exemplar models required to construct abstractions at retrieval might be expected to produce judgments more slowly than models that construct these abstractions at encoding. But consider the wide variety of processing architectures and operations available for use in conjunction with a particular representation. If we assume that an exemplar model has unlimited capacity for computing abstractions in parallel, then distinguishing this model from one that computes abstractions at encoding may be difficult, if not impossible. It would not be at all surprising if exemplar and abstraction models turn out to be indistinguishable with respect to reaction time as well. Without knowing the entire space of processing architectures and operations, this conjecture is impossible to prove deductively. But given previous experience with indistinguishability in other areas of cognitive science, the induction that reaction time will not be diagnostic is reasonable.

**Equivalence Following Information Loss**

If we discover that humans do lose information in certain ways, exemplar and abstraction models can both represent the loss observed. Because a form of each model can start with all the original information, any information loss in one can
be mimicked in the other. For example, imagine that people experience exemplars, each having four properties, and integrate this information into an ideal abstracted representation. Further imagine that, after increasing delays, people lose information about 4-tuples, then 3-tuples, and then 2-tuples, until they are only left with information about 1-tuples. An ideal exemplar model can account for this by decomposing the original exemplars into 3-tuples after some delay, by then decomposing the 3-tuples into 2-tuples after a longer delay, and by then decomposing the 2-tuples into 1-tuples after further delay. Or imagine that properties have a random probability of being deleted from exemplar representations over time. An abstraction model can mimic this by randomly lowering the frequency count of \( n \)-tuples by 1 in an appropriate manner. For example, randomly deleting a from an \( abg \) exemplar could be mimicked by randomly picking a 3-tuple (which turns out to be \( abg \)), randomly picking one of its properties (which turns out to be \( a \)), and then lowering the frequencies of the \( a, ab, ag, bg, \) and \( abg \) \( n \)-tuples by 1.

The Role of Process in Revealing Representation

As described earlier, we can never observe a representation directly. Instead we can only observe a representation through processes that operate on it. As stated by Douglas Medin, “There are no free peeks at representation” (personal communication, January 1989). It is useful to reevaluate ideal exemplar and abstraction models in this regard. Even though a model of either type stores all original exemplar information, the information that controls later performance depends on the model’s processing assumptions. To see this more concretely, consider six possible processing environments in which an ideal exemplar representation could exist (similar possibilities exist for abstraction).

**Unique Access.** An exemplar being categorized retrieves all previous memories of that exemplar and only memories of that exemplar. This processing environment insures perfect memory of exemplar information and may provide a good “peek” at representation. However, it has major limitations, such as not allowing categorization of novel exemplars (i.e., a novel exemplar retrieves no memories). Furthermore, extensive data on poor exemplar recognition cast much doubt on unique access (cf. Medin, 1986).

**Partial Matching.** An exemplar being categorized retrieves, not only previous memories of itself, but also memories of similar exemplars. Similar memories are retrieved because they partially match the exemplar being categorized. Partial matching has benefits, such as enabling the classification of novel exemplars (e.g., a novel exemplar is classified by analogy with the most similar exemplars in memory). But it also has costs, such as making it difficult to determine whether an exemplar is old or new (e.g., a novel exemplar similar to many old exemplars may retrieve many exemplar memories, making it seem old).

2. EXEMPLAR MEMORY AND ABSTRACTION

**Partial Retrieval.** An exemplar being categorized only retrieves fragments of exemplars. The missing parts of exemplars remain in storage but are inaccessible, due to factors that produce retrieval failure (e.g., Crowder, 1976). As a result, we only obtain a partial “peek” at exemplar information.

**Reconstruction.** An exemplar being categorized causes exemplar memories to be reconstructed on the basis of other knowledge in memory, such that retrieved memories contain information not originally stored in them (e.g., Bartlett, 1932). Reconstruction thereby produces a distorted “peek” at exemplar information in memory.

**Retrieval Failure.** An exemplar being categorized fails to activate previous memories of itself. These memories remain in storage but are inaccessible due to retrieval failure. As a result, the currently encoded exemplar fails to provide a “peek” at memory.

**Abstraction at Retrieval.** An exemplar being categorized causes an abstraction of several exemplars to be constructed. Instead of being utilized as individual units, exemplar memories are transformed into a single abstraction that does not correspond completely to any one exemplar (e.g., Hintzman, 1986; Kahneman & Miller, 1986). As a result, our “peeks” at exemplar information produce abstractions.

To the extent that a processing environment exhibits partial matching, partial retrieval, reconstruction, retrieval failure, and abstraction at retrieval, the output of a model increasingly diverges from information in memory. Because most cognitive theorists assume the presence of such mechanisms in humans, we face much difficulty in identifying representations. We can only observe information in a representation if cognitive processes allow us to see it. Moreover, we only observe that information in a form produced by those processes. All category information ever encountered may be stored in memory. But much of it may be unobservable or transformed, given the processing environment.

This state of affairs compounds the difficulty of determining whether people store category knowledge as exemplar memories or as abstractions. Information could be stored in one form, but the processing environment could filter, distort, and transform it so much that identifying the underlying representation is impossible.

**GENERAL CLASSES OF EXEMPLAR AND ABSTRACTION MODELS**

The ideal models described in the previous section provide two extreme forms of category learning. However, a wide variety of additional models exist, as defined by the dimensions of information storage described earlier (i.e., information
duplication, revision, and loss). Four general classes constitute the most useful parts of this space: permanent trace models, revisable trace models, cumulative abstraction models, and reductive abstraction models. As we shall see, the overlapping characteristics of the representations in these models make it difficult to sharply distinguish exemplar and abstracted representations. Moreover, given the tremendous variety of possible processing assumptions, identifying which representation underlies human performance is indeed difficult if not impossible.4

Permanent Trace Models

One class of exemplar models assumes that every processing episode stores a memory trace, which does not necessarily have to be a perfect record of the objective event. These models further assume that once a memory trace is established, it is never revised. Instead each subsequent processing of the trace, or of the exemplar that produced it, results in an additional trace but leaves the original trace intact. Loftus and Loftus (1980) found that 84% of the psychologists they sampled held this view. As discussed by Nosofsky (1988), one possible interpretation of the exemplar representation in Medin and Schaffer's (1978) context model is as permanent traces. All of the models in Estes (1986) assume permanent traces (cf. p. 504). The presence of permanent traces remains the object of much debate and research (e.g., Bekerian & Bowers, 1983; McCloskey & Zaragoza, 1985).

In general, permanent trace models store each exemplar as an independent representation in long-term memory. If the same property occurs in multiple exemplars, duplications of it are distributed across exemplar traces. Although traces are retrieved to support processing, they are not revised and therefore do not suffer intentional information loss. If abstractions are required at some point in processing, exemplar traces can be retrieved to support the necessary computations. In exemplar-only models, these abstractions are not stored. In mixed models, they are stored such that long-term memory contains both exemplars and abstractions. Abstractions in these models may then be treated similarly to exemplars (i.e., they may be duplicated and never revised); or they may be treated as in abstraction models (i.e., they may become centralized and heavily revised).

An important variant of permanent trace models assumes that exemplar traces are subject to incidental information loss. Incidental processes such as interference and decay may cause information about entire exemplars or parts of exemplars to degenerate. Such processes may similarly decrease the accessibility of exemplars or parts of exemplars, even though the information is not completely lost (Hintzman, 1986).

Revisable Trace Models

In revisable trace models, the contents of an exemplar trace may be intentionally modified by adding and/or deleting properties. Once the trace is returned to memory, information about the original exemplar is lost (e.g., Loftus, 1975; Loftus & Loftus, 1980). Revising a trace in this manner is similar to updating centralized properties in an abstraction: When information is revised, its form prior to revision no longer exists. If both the prior and revised forms are stored, then the model is a permanent trace model.

Like permanent trace models, revisable trace models exhibit information duplication, because a property may be duplicated across traces. But like abstracted representations, revisable traces exhibit intentional information revision. With sufficient processing, a frequently revised trace could become a centralized information structure, much like an abstraction.

Cumulative Abstraction Models

Cumulative abstraction models generally exhibit revision of nonduplicated information: If the same property or combination of properties occurs across exemplars, centralized property information is revised. More specifically, frequencies of n-tuples are revised after each new exemplar, such that frequency information accumulates for properties and/or property combinations. Within this general class of models, many variations are possible. As described earlier, cumulative models are ideal if they store complete information about property cooccurrence (e.g., A4 in Fig. 2.3). More specifically, ideal cumulative models store the highest-order n-tuples available—what will be referred to as cardinal n-tuples. For example, if all exemplars have three properties, then the cardinal n-tuples are 3-tuples. If cardinal n-tuples are stored and distinguishable, all exemplar information is available.5

Perhaps the best known model that uses cumulative abstraction is the modal prototype model. This model is the one most often tested and rejected by exemplar theorists (e.g., Medin & Schaffer, 1978). It assumes that an exemplar is represented by a set of attribute values, such as values for color, shape, and size. For example, an exemplar might be green, oval, and small. Frequencies of values across exemplars are maintained, at least initially. However, the abstraction (prototype) that is eventually produced and that controls all subsequent processing contains only the modal value for each attribute. Consequently, modal pro-

4By no means is this the only way to taxonomize models of category learning (cf. Estes, 1986; Hayes-Roth & Hayes-Roth, 1977; Reed, 1972; Smith & Medin, 1981).

5In the literature, cumulative abstraction models are often referred to as "frequency" and "power set" models. Interestingly, some forms of these abstraction models exhibit information duplication, contrary to my definition of abstracted representation (e.g., Hayes-Roth & Hayes-Roth, 1977; Reitman & Bower, 1973). In these cases, a property is often duplicated in different n-tuples (e.g., a in the n-tuples for ab, abc, and ace). This is one of several cases in this section where the distinction between exemplar and abstracted representations becomes blurred.
prototype models initially track frequency information for all 1-tuples but then discard most of it, once the most frequent value for each attribute is identified. In addition, modal prototype models don’t record cooccurrence information. The clear rejection of these models in recent studies demonstrates that people do not discard all idiosyncratic and cooccurrence information. As we have seen, however, models that use other abstracted representations, such as $A_4$, do record these two types of information and are therefore not rejected by these findings.

A wide variety of cumulative abstraction models can be gradually constructed by adding additional assumptions to modal prototype models. First, models can store increasing amounts of information about independent property frequency. The frequencies of all characteristic properties—not just modal properties—could be stored, as could the frequencies of all idiosyncratic properties. Second, models can store increasing amounts of cooccurrence information. Minimally, pairwise cooccurrence could be included, as represented by 2-tuple frequency (e.g., $A_3$ in Fig. 2.4). Frequencies of higher-order $n$-tuples could be added gradually to construct models that store greater amounts of cooccurrence information (e.g., $A_4$ in Fig. 2.4). Once the storage of cardinal $n$-tuples occurs, the abstraction becomes ideal.

An interesting model—the cardinal cumulative model—only stores the frequencies of cardinal $n$-tuples and not the frequencies of lower-order $n$-tuples. For example, if all exemplars have three properties, then only information about 3-tuples is stored—not information about 1-tuples or 2-tuples. This model corresponds to the right side of $A_4$ in Fig. 2.4, after removing the information about 2-tuples and 1-tuples. This model only abstracts at the level of exemplars—abstractions for 1-tuples and 2-tuples are not computed. Of course, one might wish to add a processor that could compute additional abstractions if needed later. Because all information about original exemplars is stored, cardinal cumulative models are ideal. They are also much more storage efficient than other ideal cumulative models, such as $A_4$.

Cumulative abstraction models, like exemplar models, may also lose information due to incidental degeneration. Incidental processes such as interference and decay may cause entire $n$-tuples or parts of $n$-tuples to degenerate. Such processes may similarly decrease the accessibility of $n$-tuples, or parts of $n$-tuples, even though the information is not completely lost.

Reductive Abstraction Models

Consider the revision of cumulative abstractions. Revision basically involves adding frequency to $n$-tuples. As each new exemplar is encoded, every relevant $n$-tuple being tracked is augmented by 1. Consequently, the revision of $n$-tuples does not lose information. Because each unit of frequency represents an exemplar, information about that $n$-tuple in the original exemplars can be reconstructed. Revision of an $n$-tuple loses none of the original exemplar information about it.

In contrast, the revision of property information produces information loss in reductive abstraction models. For a given $n$-tuple, information across exemplars is combined in some way that produces useful category information but that loses information about exemplars. Perhaps the classic case is the average distance model. One way to view this model is in terms of a multidimensional space.
Imagine a category of colored circles varying on the dimensions of size, brightness, and hue. Each exemplar can be viewed as defining a point (vector) in the three-dimensional space defined by these dimensions. According to the average distance model, only the average vector across exemplars represents the category, namely, the point in the space that represents the average size, brightness, and hue of the circles. By computing the average vector and discarding exemplar information, this model loses information about the original exemplars. Of course, a model could store every exemplar as a point in the space, but then this would be an ideal exemplar model.

Each value on a dimension can be viewed as a 1-tuple (e.g., blue is a 1-tuple for color). A prototype can therefore be viewed as those 1-tuples that represent average values across exemplars on relevant dimensions. When a new exemplar is encoded, the revision that takes place on each average 1-tuple is to move it along its dimension toward the exemplar’s value for that dimension. In other words, each new exemplar is averaged into the current prototype.

This type of revision differs fundamentally from the revision that occurs in cumulative abstraction models. In cumulative models, the revision of a 1-tuple simultaneously preserves information about individual exemplars while establishing overall information about the category. Information about the number of exemplars containing the 1-tuple is always available, because each unit of frequency represents an exemplar. In contrast, averaging 1-tuples loses information about the number of exemplars sharing a given 1-tuple. All that remains is the average. In this regard, average distance models produce reductive abstractions, because revision reduces units of exemplar information to some more distilled form.

Theorists have proposed a wide variety of models that produce reductive abstractions (e.g., Estes, 1986; Hayes-Roth & Hayes-Roth, 1977; Reed, 1972; Smith & Medin, 1981). For the most part, these models either reduce information by transforming frequency information into averages or probabilities (e.g., cue validities). In all cases, information is lost through the transformation of frequency units.

**Connectionist Models.** In nearly all previous cases, reductive abstraction models only track 1-tuples, failing to record cooccurrence information. These models typically do not compute averages or probabilities for higher-order n-tuples. What distinguishes connectionist models from previous reductive models is that connectionist models store cooccurrence information. But in the process of reducing cooccurrence information, connectionist models lose information, which distinguishes them from cumulative models that track cooccurrence.

Figure 2.4 shows how connectionist models store cooccurrence information but reduce it in the process. To see this, first consider A₅, which represents exemplars e₅ through e₁₀ in Fig. 2.2. A₅ is a cumulative abstraction that contains information about 1-tuples and 2-tuples. Information about 3-tuples, and therefore about the original exemplars, has been lost. Now consider representation C₁ in Fig. 2.4, which is a simple connectionist representation. The pluses and minuses to the left represent positive and negative correlations between pairs of properties. The pluses below the properties represent positive correlations between the properties and the category label (not shown).

Like A₅, C₁ tracks information about 1-tuples (correlations between a property and the category) and 2-tuples (correlations between pairs of properties). But whereas A₅ contains complete information about these n-tuples, C₁ has lost information about their absolute frequency. C₁ reduces frequency information to update correlations between pairs of properties, representing these correlations as weights. To the extent that one property is diagnostic of another, their weight becomes positive. To the extent that one property predicts the absence of another, their weight becomes negative.

Next consider representations A₄ and C₂ in Fig. 2.4. A₄ is the ideal cumulative abstraction discussed earlier. Information about every n-tuple has been recorded, and no information has been lost. C₂ is a connectionist representation with hidden units h₁ through h₄. Hidden units allow a connectionist model to store information about complex correlations between properties. Whereas connectionist models without hidden units can only store information about 2-tuples, connectionist models with hidden units can store information about 3-tuples, 4-tuples, etc., depending on the number of initial units that hidden units integrate.

A connectionist model with sufficient hidden units can record correlations for all possible n-tuples across exemplars, analogous to an ideal cumulative model. However, these connectionist models still lose information. Because the weights that represent n-tuples are essentially correlations, it is impossible to reconstruct the original exemplars. Such reconstruction is possible only if absolute frequencies are preserved, as in ideal cumulative models. Connectionist models also lose information through incidental degeneration. Interference and decay typically drive the weights in these models toward zero over time and experience.

Perhaps most unique to connectionist models are the learning rules they use to integrate exemplar information into connection weights (see Gluck & Bower, 1988, for discussion). These rules go far beyond the methods of reduction previously employed in abstraction models. Connectionist models also represent

---

6To simplify presentation, inhibitory links between properties and hidden units are not shown in C₂. However, the reader should assume that paths with negative weights connect all properties and hidden units that are not shown connected.

7Connectionist models may also lose information in a more severe manner when representing multiple categories. Imagine that the correlation between properties a and h is positive for category X but equally negative for category Y. Across categories, the correlation is therefore 0, assuming equally sized categories and comparable learning sequences. Nonoptimal information about the correlation with respect to each category is stored because the same correlational weight, 0, is used for both categories. This problem can be eliminated by using hidden units that associate particular values of a correlation with particular categories. For example, hidden units that represent a positive correlation between a and h could be associated with category X, whereas hidden units that represent a negative correlation could be associated with category Y. A similar problem exists for cumulative abstraction models, in that a given n-tuple needs to be indexed for each category in which it occurs.
negative correlations between properties explicitly, whereas previous abstraction models have not. However, cumulative abstraction models could probably utilize connectionist learning rules in some form, and they contain the requisite information for producing information about negative correlations.

An interesting limit on connectionist representations may be a difficulty in supporting frequency estimates (e.g., Hasher & Zacks, 1979; Hintzman, 1976). Although connectionist models readily support probability estimates (Gluck & Bower, 1988), it is not clear how they would support estimates of absolute frequency (J. Jonides, personal communication, November, 1988). Moreover, an initial attempt to represent absolute frequency in a connectionist model has met with difficulty (F. T. Durso & T. Dayton, personal communication, November 1988). Because connectionist models reduce frequency information to obtain correlational weights, these models lose information about how often exemplars and n-tuples occur. Traditional reductive representations, such as in the average distance model, also have difficulty producing frequency estimates, given all exemplar information is lost in the process of computing averages or probabilities. In contrast, exemplar representations and cumulative abstractions readily support frequency estimates of how many times a particular exemplar occurred and how many exemplars possessed a particular property.

DEVELOPING THEORIES OF CATEGORY LEARNING

If we cannot determine whether people represent categories with exemplars or abstractions, then what have we learned from years of research on category learning? Clearly we have developed an impressive store of empirical facts about category learning. Some of these are of substantial importance in understanding human knowledge. For example, people use idiosyncratic information about exemplars in categorization decisions; people use cooccurrence information in categorization decisions; people do not represent categories with static representations but instead represent them dynamically. Certainly many other important findings have been established as well (for reviews, see Mervis & Rosch, 1981; Medin & Smith, 1984; Oden, 1987; Smith & Medin, 1981).

As I have argued, we can not say whether category knowledge is distributed in exemplars or centralized in abstractions. But we do know that any account of knowledge that excludes idiosyncratic information, cooccurrence information, or dynamic representation is inadequate. Consequently, the classical theories of knowledge that have dominated philosophy, linguistics, psychology, and computer science for years are probably wrong (Smith & Medin, 1981). Many theorists now believe that the large body of evidence developed from the study of category learning has clearly rejected this position. Similarly, we have rejected large classes of exemplar and abstraction models that are insensitive to idiosyncratic and cooccurrence information (Medin & Schaffer, 1978).

How should we proceed in developing models of category learning? If we can develop models with either exemplar or abstracted representations, which should we chose? One tack is to pursue models that stimulate new research. At one time, prototype models were highly stimulating in this regard, generating numerous lines of productive inquiry. More recently, exemplar models have played this role. We should certainly encourage any model that inspires empirical and theoretical progress.

A second tack is to favor models whose assumptions seem most plausible. As noted by Lee Brooks, "the game is to produce a sufficiently peculiar pattern of data" such that the "opposition is forced into considerable contortions to accommodate it," where "the special assumptions they have to invent do no other useful work . . . in the current research context" (personal communication, January 1989). Because exemplar and abstraction models both have sizable followings, it is not clear that either model has an advantage in this regard. Perhaps future research will produce a greater consensus.

A third tack is to develop the space of category learning models (e.g., Estes, 1986). Formal analyses are often productive in identifying new models of interest and in determining whether models are empirically distinguishable. Much development of learning models remains to be done. Hopefully, such development will direct effort toward issues that are resolvable and away from those that are not.

A fourth tack is to look for guidance outside the category learning literature. One possibility is theory from other areas of cognitive psychology. What assumptions about representation best serve perception, attention, memory, language, problem solving, and reasoning? Findings and theory from these areas may suggest one type of representation over another. A second source of outside constraint is computational efficiency. From implementing different representations on various types of artificial hardware, we may find that one type of representation is computationally more tractable than another. A third source of outside constraint in neuroscience. At some point, we may develop neurological evidence for particular forms of representation.

Clearly, numerous possibilities exist for future research. As I have argued here, however, one must observe caution in making claims about representation. Because exemplar and abstracted representations can potentially represent the same information, and because we can only observe representation through processing, concluding that people use one representation and not another is difficult, if not impossible. Instead our empirical efforts can only support and reject particular models, namely, representation-process pairs.

ACKNOWLEDGEMENTS

Work on this manuscript was supported by National Science Foundation Grant IRI-8609187 and Army Research Institute Contract MD A903-86-C-0172. I am particularly grateful to Douglas Medin for ongoing discussion of this topic and
for extensive comments on previous drafts, all of which have had a major impact on this paper. I am also grateful to Brian Ross, Lee Brooks, Mark Gluck, Joel Martin, and David Meyer for helpful comments. None of these colleagues necessarily agrees with any particular point herein. Nor are they responsible for any errors that may remain. Work on this paper was performed while the author was a visitor at the University of Michigan. Address correspondence to Lawrence W. Barsalou, School of Psychology, Georgia Institute of Technology, Atlanta, GA 30332.

REFERENCES


Smith makes the interesting suggestion that cognitive processes have signatures that can be used for their positive identification. This is a seductive idea, and Smith goes a long way towards demonstrating its validity. Still, going a long way is not the same as going all the way, especially when seduction is at issue. The gist of my commentary is that it is premature to speak of cognitive signatures in social cognition because the results we strive to explain are usually open to several alternative explanations.

At the heart of Smith’s paper is the methodological principle of specificity, which focuses on the effects of prior experiences on later performance. The argument is that cognitive mediators can be identified by their pattern of specificity on a number of dimensions (content, process, context and time). Three types of cognitive mediators receive particular attention in the paper, those based on: (a) abstract knowledge structures, (b) exemplars, and (c) procedures. The thrust of Smith’s paper is that theorists in social cognition may have paid far too much heed to the power of abstract knowledge structures in explaining social phenomena. Instead, the argument goes, phenomena that have been attributed to abstract knowledge structures in the past can be accommodated by exemplar-based and procedural representations. Moreover, the patterns of specificity revealed by relevant data seem to bear the signatures of these latter mediators rather than that of the former.

My commentary focuses on the parts of Smith’s argument having to do with the role of productions in social cognition. Because I have recently been working on a model of the relationship between implicit and explicit memory that is relevant to data collected by Smith and Branscombe (1988), my discussion is organized around the interpretation of these data.