COGNITIVE SCIENCE 19, 141-205 (1994)

MAC/FAC: A Model of Similarity-based Retrieval

KENNETH D. FORBUS,
DEDRE GENTNER,
KEITH LAW
Northwestern University

We present a model of similarity-based retrieval that attempts to capture three seemingly contradictory psychological phenomena: (a) structural commonalities are weighed more heavily than surface commonalities in similarity judgments for items in working memory; (b) in retrieval, superficial similarity is more important than structural similarity; and yet (c) purely structural (analogical) remindings e sometimes experienced. Our model, MAC/FAC, explains these phenomena in terms of a two-stage process. The first stage uses a computationally cheap, non-structural matcher to filter candidate long-term memory items. It uses *content vectors*, a redundant encoding of structured representations whose dot product estimates how well the corresponding structural representations will match. The second stage uses SME (structure-mapping engine) to compute structural matches on the handful of items found by the first stage. We show the utility of the MAC/FAC model through a series of computational experiments: (a) We demonstrate that MAC/FAC can model patterns of access found in psychological data; (b) we argue via sensitivity analyses that these simulation results rely on the theory; and (c) we compare the performance of MAC/FAC with ARCS, an alternate model of similarity-based retrieval, and demonstrate that MAC/FAC explains the data better than ARCS. Finally, we discuss limitations and possible extensions of the model, relationships with other recent retrieval models, and place MAC/FAC in the context of other recent work on the nature of similarity.

1. INTRODUCTION

Similarity-based remindings range from the sublime to the stupid. At one extreme, seeing the periodic table of elements reminds one of octaves in music. At the other, a bicycle reminds one of a pair of eyeglasses. Often,

This research was supported by the Office of Naval Research (Contract No. N00014-89-J-1272). We thank Ray Bareiss, Mark Burstein, Gregg Collins, Ron Ferguson, Brian Falkenhainer, Rob Goldstone, Art Markman, Doug Medin, and Mary Jo Rattermann for discussions of these issues. We thank Paul Thagard for providing us with ARCS and its associated databases.

Correspondence and requests for reprints should be sent to Kenneth D. Forbus, Northwestern University, The Institute for the Learning Sciences, 1890 Maple Avenue, Evanston, IL 60201.

remindings are neither brilliant nor superficial but simply mundane, as when a bicycle reminds one of another bicycle. Theoretical attention is inevitably drawn to spontaneous analogy: That is, to structural similarity unsupported by surface similarity, as in the octave/periodic table comparison. Such remindings seem clearly insightful and seem linked to the creative process and should be included in any model of retrival. But, as we review below, research on the psychology of memory retrieval points to a preponderance of the latter two types of similarity: (mundane) literal similarity, based on both structural and superficial commonalities; and (dumb) superficial similarity, based on surface commonalities. A major challenge for research on similarity-based remindings is to devise a model that will produce chiefly literal similarity and superficial remindings, but still produce occasional analogical remindings.

A further constraint on models of access comes from considering the role of similarity in transfer and inference. The large number of superficial remindings indicates that retrieval is not very sensitive to structural soundness. But appropriate transfer requires structural soundness, so that knowledge can be exported from one description into another. And psychological evidence (also discussed below) indicates that the mapping process involved in transfer is actually very sensitive to structural soundness. Hence our memories often give us information we don't want, which at first seems somewhat paradoxical. Any model of retrieval should explain this paradox.

This article presents MAC/FAC, a model of similarity-based reminding that attempts to capture these phenomena. MAC/FAC models similarity-based retrieval as a two-stage process. The first stage (MAC) uses a cheap, nonstructural matcher to quickly filter potentially relevant items from a pool of such items. These potential matches are then processed in the FAC stage by a more powerful (but more sensitive) structural matcher, based on the structure-mapping notion of literal similarity (Gentner, 1983).

We begin in Section 2 by briefly reviewing psychological evidence on similarity-based retrieval and mapping, thereby extracting some criteria which retrieval models must satisfy. This section also outlines the computational issues raised by similarity-based retrieval, drawing on the AI literature as necessary. Section 3 describes the MAC/FAC model, showing how it satisfies the psychological and computational desiderata. Section 4 illustrates the model's psychological plausibility by simulating the results of a psychological experiment. Section 5 explores the consequences of different design decisions by sensitivity analyses at the level of algorithms, demonstrating that the model's performance depends on the theoretically important parameters. Section 6 compares MAC/FAC with ARCS, the closest competing model of similarity-based retrieval, demonstrating that MAC/FAC performs well on databases designed by others (e.g., the ARCS data sets) and that MAC/FAC's performance fits the psychological evidence better than

ARCS. Finally, Section 7 compares MAC/FAC to several other memory models, analyzes some of its limitations, and discusses possible extensions.

2. FRAMEWORK

Similarity-based transfer can be decomposed into subprocesses. Given that a person has some current *target* situation in working memory, transfer from prior knowledge requires at least

1. *accessing* a similar (*base*) situation in long-term memory,
2. *creating a mapping* from the base to the target, and
3. *evaluating* the mapping.

In this case, the base is an item from memory, and the target is the probe; that is, we think of the retrieved memory items as mapped to the probe. Other processes may also occur—*verifying new inferences* about the target (Clement, 1986), *elaborating* the base and target (Falkenhainer, 1988; Ross, 1987), *adapting* or *tweaking* the domain representations to improve the match (Falkenhainer, 1990a, b; Holyoak, Novick, & Melz, 1994; Kass, 1986, 1989), and *abstracting* the common structure from base and target (Gick & Holyoak, 1983; Skorstad, Gentner, & Medin, 1988; Winston, 1982) —but our focus is on the first three processes.

2.1 Structure-Mapping and the Typology of Similarity

The process of *mapping* aligns two representations and uses this alignment to generate analogical inferences (Gentner, 1983, 1988, 1989b). Alignment occurs via matching, which creates correspondences between items in the two representations. Analogical inferences are generated by using the correspondences to import knowledge from the base representation into the target. The mapping process is assumed to be governed by the constraints of *structural consistency: one-to-one mapping* and *parallel connectivity. One-to-one mapping* means that an interpretation of a comparison cannot align (e.g., place into correspondence) the same item in the base with multiple items in the target, or vice versa. *Parallel connectivity* means that if an interpretation of a comparison aligns two statements, their arguments must also be placed into correspondence.[1] In this account, similarity is defined in terms of correspondences between structured representations (Gentner, 1983; Gentner & Markman, 1993, 1994a, 1994b; Goldstone & Medin, 1994a, 1994b; Goldstone, Medin, & Gentner, 1991; Markman & Gentner, 1990, 1993a, 1993b; Medin, Goldstone, & Gentner, 1993). Matches can be distinguished according to the kinds of commonalities present. An *analogy* is a match based on a common system of relations, especially involving higher-

[1] Previously we used the term *structurally grounded* for parallel connectivity.

order relations.[2] A *literal-similarity* match includes both common relational structure and common object descriptions. A *surface similarity* or *mere-appearance* match is based primarily on common object descriptions, with perhaps a few shared first-order relations.

There is considerable evidence that the mapping process is sensitive to structural commonalities. People can readily align two situations, preserving structurally important commonalities, making the appropriate lower-order substitutions, and mapping additional predicates into the target as *candidate inferences*. For example, Clement and Gentner (1991) showed people analogies and asked which of two lower-order assertions, both shared by base and target, was most important to the match. Subjects chose assertions that were connected to matching causal antecedents: That is, their choice was based not only on the goodness of the local match but also on whether it was connected to a larger matching system. In a second study, subjects were asked to make a new prediction about the target based on the analogy with the base story. They again showed sensitivity to connectivity and systematicity in choosing which predicates to map as candidate inferences from base to target. Evidence for structural consistency in mapping comes from a study by Spellman and Holyoak (1992). They asked people to explicate the analogy between the Gulf War and World War II, assuming Saddam Hussein maps onto Hitler. Although people were divided in their mappings, they were highly consistent. People who mapped Bush onto Churchill mapped the current USA onto World War II Britain, and people who mapped Bush onto F.D.R. mapped the USA today onto the USA during World War II.

The degree of relational match is also important in determining people's evaluations of comparisons. People rate metaphors as more apt when they are based on relational commonalities than when they are based on common object descriptions (Gentner, 1988; Gentner & Clement, 1988). Gentner, Rattermann, and Forbus (1993) asked subjects to rate the soundness and similarity of story pairs that varied in which kinds of commonalities they shared. Subjects' soundness and similarity ratings were substantially greater for pairs that shared higher-order relational structure than for those that did not (Gentner & Landers, 1985; Gentner, Rattermann, & Forbus, 1993; Rattermann & Gentner, 1987). Common relational structure also contributes strongly to judgments of perceptual similarity (Goldstone et al., 1991) as well as to the way in which people align pairs of pictures in a mapping task (Markman & Gentner, 1990, 1993b) and determine common and distinctive features (Gentner & Markman, 1994a, b; Markman & Gentner, 1993a).

[2] We define the *order* of an item in a representation as follows: Objects and constants are order 0; the order of a statement is 1 plus the maximum of the order of its arguments.

Any model of human similarity and analogy must capture this sensitivity to structural commonality. To do so, it must involve structural representations and processes that operate to align them (Barnden, 1994; Gentner & Markman, 1994a, b; Goldstone et al., 1991; Holyoak et al., 1994; Keane, 1988a, 1988b; Markman & Gentner, 1993a, 1993b; Medin et al., 1993; Reed, 1987; Reeves & Weisberg, 1994). This would seem to require abandoning some highly influential models of similarity: for example, modeling similarity as the intersection of independent feature sets or as the dot product of feature vectors. However, we will show that a variant of these nonstructural models can be useful in describing memory retrieval.

2.1.1 Similarity-based Access from Long-term Memory

There is considerable evidence that access to long-term memory relies more on surface commonalities and less on structural commonalities than does mapping. For example, people often fail to access potentially useful analogs, as in Gick and Holyoak's (1980, 1983) dramatic demonstration. When subjects were told a story and then given an analogous problem to solve, about 30% solved the problem. However, if subjects were simply told to think about the story they had heard, 80% to 90% solved the problem. We can infer that most of the subjects retained representations of the prior story sufficient to provide a useful analogy, but that hearing the structurally analogous problem did not provide spontaneous access to the story representation in memory. Other research has shown that, although people in a problem-solving task are often reminded of prior problems, these remindings are often based on surface similarity rather than on structural similarities between the solution principles (Holyoak & Koh, 1987; Keane, 1987, 1988b; Novick, 1988a, b; Reed, Ernst, & Banerji, 1974; Ross, 1984, 1987, 1989; see also the comprehensive review by Reeves & Weisberg, 1994).

The experiments we will model here investigated which kinds of similarities led to the best retrieval from long-term memory (Gentner & Landers, 1985; Gentner, Rattermann, & Forbus, 1993; Rattermann & Gentner, 1987). Subjects were first given a relatively large memory set (the "Karla the Hawk" stories). About a week later, they were given new stories that resembled the original stories in various ways and were asked to write out any remindings they experienced to the prior stories while reading the new stories. Finally, they rated all the pairs for soundness—that is, how well inferences could be carried from one story to the other. The results showed a marked disassociation between retrieval and subjective soundness and similarity. Surface similarity was the best predictor of memory access, and structural similarity was the best predictor of subjective soundness. This dissociation held not only between subjects but also within subjects. That is, subjects given the soundness task immediately after the cued retrieval task judged that the very matches that had come to their minds most easily (the surface matches) were highly unsound (i.e., unlikely to be useful in inference). This suggests that similarity-based access may be based on qualitatively distinct processes from analogical inferencing.

It is not the case that higher-order relations contribute nothing to retrieval. Adding higher-order relations led to nonsignificantly more retrieval in two studies and to a small but significant benefit in the third. Other research has shown positive effects of higher-order relational matches on retrieval, especially in cases where subjects were brought to do intensive encoding of the original materials (Faries & Reiser, 1988) or were expert in the domain (Novick, 1988a, 1988b). But higher-order commonalities have a much bigger effect on mapping once the two analogs are present than they do on similarity-based retrieval, and the reverse is true for surface commonalities.

These results place several constraints on a computational model similarity-based retrieval. The first two criteria ensure that the model can provide an account of mapping and inference:

> *Structured representation criterion:* The model must be able to store structured representations.
>
> *Structured mappings criterion:* The model must incorporate processes of structural mapping (i.e., alignment and transfer) over its representations.

The remaining four criteria summarize the pattern of retrieval results:

> *Primacy of the mundane criterion:* The majority of retrievals should be literal similarity matches: that is, matches high in both structural and surface commonalities.
>
> *Surface superiority criterion:* Retrievals based on surface similarity are frequent.
>
> *Rare insights criterion:* Relational remindings must occur at least occasionally, with lower frequency than literal similarity or surface remindings.
>
> *Scalability criterion:* The model must be plausibly capable of being extended to large memory sizes.

No current model of transfer succeeds in satisfying all six criteria. There are two major approaches to memory models: indexing models, commonly used in case-based reasoning work, and feature-vector models, commonly used in mathematical modeling of human memory. We examine the trade-offs of each in turn.

Most case-based reasoning models (Birnbaum & Collins, 1989; Branting, in press; Kass, 1986, 1989; Kolodner, 1984, 1988, 1989, 1993; Schank, 1982) use structured representations and focus on the process of adapting and applying old cases to new situations. Such models satisfy the structured representation and structured mappings criteria. However, such models also typically presume a highly indexed memory in which the vocabulary used for indexing captures significant higher-order abstractions such as themes and principles. Viewed as psychological accounts, these models would predict that people should typically access the best structural match. That prediction fails to match the pattern of psychological results summarized by the primacy of the mundane and surface superiority criteria. Scalability is also an open question at this time, because no one has yet accumulated and indexed a large (say 10^3 to 10^6) corpus of structured representations.

The reverse set of advantages and disadvantages holds for approaches that model similarity as the result of a dot product (or some other simple operation) over feature vectors, as in many mathematical models of human memory (e.g., Gillund & Shiffrin, 1984; Hintzman, 1986, 1988; Medin & Schaffer, 1978; but see Murphy & Medin, 1985) as well as in many connectionist models of learning (e.g., Smolensky, 1988; see also reviews by Humphreys, Bain, & Pike, 1989, and Ratcliff, 1990). These models typically use nonstructured knowledge representations and relatively simple match processes and hence do not allow for structural matching and inference. Such models also tend to use a unitary notion of similarity, an assumption that is called into question by the dissociation described earlier (see also Gentner & Markman, 1993; Medin et al., 1993). However, the use of feature vectors has some advantages for modeling access to long-term memory. The computations are simple enough to make it feasible to compute many matches and choose the best, thus satisfying the scalability criterion. Furthermore, because object features are included in the feature vectors, these models should be able to capture the surface superiority criterion and in many cases the primacy of the mundane criterion. (Failures on the latter will occur for cross-mappings, when the objects and relations match but their bindings do not.) It should be noted that some case-based reasoning work also restricts itself to feature-vector representations and thus has the same strengths and weaknesses (e.g., Stanfill & Waltz, 1986).

The MAC/FAC model seeks to combine the advantages of both approaches. We turn now to its description.

3. THE MAC/FAC MODEL

The complexity of the phenomena in similarity-based access suggests a two-stage model. Consider the computational constraints on access. The large number of cases in memory and the speed of human access suggests a computationally cheap process. But the requirement of judging soundness, essential to establishing whether a match can yield useful results, suggests an expensive match process. A common computational solution to such problems is to use a two-stage process, in which a cheap filter is used to pick out a subset of likely candidates for more expensive processing (cf. King & Bareiss, 1989; Waltz, 1989). MAC/FAC uses this strategy. The disassociation noted previously can be understood in terms of the interactions of its two stages.

Figure 1 illustrates the components of the MAC/FAC model. The inputs are a pool of memory items and a *probe*, that is, a description for which a match is to be found. The output is an item from memory (i.e., a structured description) and a comparison of this item with the probe. (Section 3.1 describes exactly what a comparison is.) Internally there are two stages. The MAC stage provides a cheap but nonstructural filter, which only passes on a handful of items. The FAC stage uses a more expensive but more accurate structural match to select the most similar item(s) from the MAC output, producing a full structural alignment. Each stage consists of *matchers*, which are applied to every input description, and a *selector*, which uses the evaluation of the matchers to select which comparisons are produced as the output of that stage. Conceptually, matchers are applied in parallel within each stage.

We make minimal assumptions concerning the global structure of long-term memory. We assume here only that there is a large pool of descriptions from which we must select one or a few that are most similar to a probe. We are uncommitted as to whether the pool is the whole of long-term memory or a subset selected via some other method, for example, spreading activation.

We begin by describing the FAC stage. In doing so, we also describe the computational framework which underlies MAC and FAC, including our conventions for representation and the information about the SME algorithm that is required to fully understand MAC/FAC.

3.1 The FAC Stage and SME

The FAC stage takes as input the descriptions selected by the MAC stage and computes a full structural match between each item and the probe. We model the FAC stage by using SME, the *structure-mapping engine* (Falkenhainer, Forbus, & Gentner, 1986, 1989). Here we briefly summarize SME's operation, both by way of describing the FAC stage and to provide the vocabulary needed to motivate the design of the MAC stage.

SME is an analogical matcher designed as a simulation of structure-mapping theory. It takes two inputs, a base description and a target description. (For simplicity we speak of these descriptions as being made up of items, meaning both objects and statements about these objects.) It computes a set of *global interpretations* of the comparison between base and target. Each global interpretation includes the following.

- A set of *correspondences* which pair specific items in the base representation to specific items in the target.
- A *structural evaluation* reflecting the estimated soundness of the match. In subsequent processing, the structural evaluation provides one source of information about how seriously to take the match.
- A set of *candidate inferences,* potential new knowledge about the target which is suggested by the correspondences between the base and target.

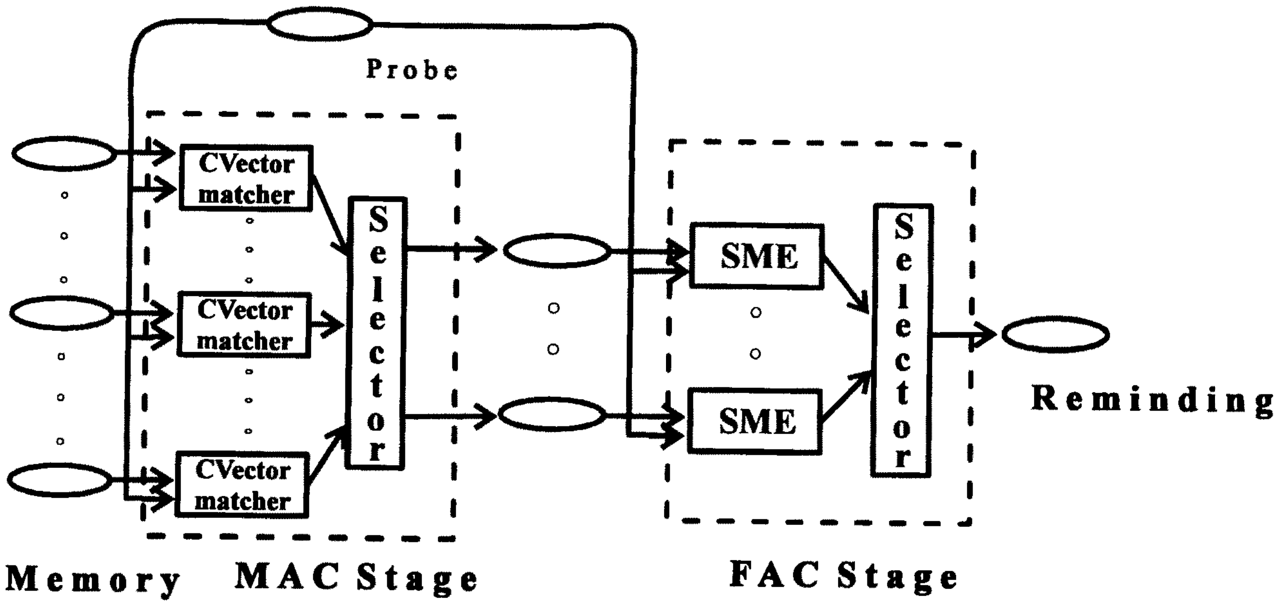


Figure 1. The MAC/FAC model.

Candidate inferences are what give analogy its generative power, because they represent the importation of new knowledge into the target description. However, they are only conjectures; they must be tested and evaluated by other means.

We can illustrate these ideas with the Rutherford analogy, which describes the structure of the atom in terms of that of the solar system. The solar system is the base description and the atom is the target description.

- Rutherford paired the Sun to the nucleus and the planets to the electrons. These correspondences seem reasonable not because of intrinsic object similarities but because they allow various relational statements also to be placed in correspondence (i.e., *aligned*): for example, the relative masses of the objects and the fact that the planets/electrons revolve around the Sun/nucleus.
- This interpretation is a selection from among many common relations. It focuses on the causal system of a central gravitational/electromagnetic force, the relative mass of the two bodies within each system, and the fact that the less massive body revolves around the heavier body. Other common relations—such as the relative temperatures or differences in color of the two objects—that do not belong to a common connected system are not included in the interpretation. We refer to this preference for connected systems of common predicates as the *systematicity* principle.
- The preferred interpretation might also sanction new conjectures about the atom, such as that the cause of the electrons revolving around the nucleus is the existence of an attractive force.[3]

The interpretations produced by SME are structurally consistent, in that they satisfy the constraints of one-to-one mapping and parallel connectivity, as defined in Section 2.1. These constraints are important because they allow for the generation of coherent candidate inferences. The systematicity constraint is important because it captures the human preference for aligning connected systems of predicates (e.g., logical arguments or causal sequences). In addition, SME attempts to find *maximal* interpretations. An interpretation is maximal if adding any additional correspondences would render it structurally inconsistent. Maximality is important both because it reduces the number of possible interpretations and because it ensures that the full structural implications of a set of correspondences will be considered.

Before describing the SME algorithm further, some conventions concerning representation are in order. We use infix notation or Lisp prefix syntax for statements as appropriate. We use the term *functor* of a statement

[3] Incorrect candidate inferences are also possible—for example, that the attractive force in the atom is gravity. What counts as a candidate inference versus an alignable (or nonalignable) structure depends on the reasoner's state of knowledge about the target.

as a general term for the relation or function or connective that takes the remaining parts of the statement as its arguments. For example,

1. In GREATER-THAN (HEIGHT (A), HEIGHT (B)), the functor is GREATER-THAN.
2. In NOT (ABOVE (B, A)), the functor is NOT.
3. In HEIGHT (A), the functor is HEIGHT.
4. In RED (A), the functor is RED.

Example #1 is an example of a *relation.* Relations range over truth values, and their arguments can be entities or other statements. Relations always have multiple arguments, with the exception of logical connectives (e.g., Example #2), which are always treated as relations regardless of the number of arguments. For the purposes of structure-mapping, modal operators and other higher-order predicates are classified as relations. Example #3 is an example of a *function*, which maps one or more entities into another entity or constant. In our psychological modeling, functions are often used to represent known dimensions or components of structured objects (e.g., height, pressure, or color). Example #4 is an example of an *attribute,* an atomic description of some property of an entity. Attributes take only one argument to capture the notion of a unitary description. This of course does not mean that attributes cannot be decomposed. For instance, the following forms are logically equivalent:

- RED (A)
- COLOR-OF (A, red)
- COLOR (A) = red

However, we use these three distinct forms to represent distinct psychological constructs. Roughly, the first, an attribute, indicates that the subject thinks of redness as a quality of the object. The second, a relation, indicates that the subject has to some degree disengaged redness from the object and sees color as a relationship between an object and a set of possible values. The third, a function, indicates that the subject conceives a color as a dimension of general application and thinks of the color of A as a value along this dimension. We view this kind of dimensional representation as important because dimensions may in the process of comparison be aligned with quite different dimensions (e.g., HEIGHT and DARKNESS). Thus, qualities that are conceived as of dimensions are more likely to participate in systematic cross-dimensional matches. (For the implications of this idea in analogical development, see Gentner & Rattermann, 1991; Gentner, Rattermann, Kotovsky, & Markman, in press; Kotovsky & Gentner, 1990.)

With these conventions in mind, let us turn to the SME algorithm. SME operates via a local-to-global process. Conceptually, its operation can be divided into four phases. The first phase constructs a network of local matches between items in the base and target. The second phase constructs global interpretations by coalescing structurally consistent combinations of local matches. The third phase computes the structural evaluation, and the fourth phase computes candidate inferences for each interpretation. We examine each in turn.

SME begins by finding all possible local matches between statements in the base and statements in the target. A local match is created between base item B_i and target item T_j when either

1. B_i and T_j are both statements whose functors are sufficiently alike (typically identical, but see below), or
2. B_i and T_j are corresponding arguments of other statements which are connected by a local match and are both either objects or functions.

For instance, given the base item B1 and the target item T1 defined as

```
B1: (CAUSE Event17 Event31)
T1: (CAUSE Event5 Event63),
```

a match would be hypothesized between B1 and T1 because their functors (i.e., CAUSE) are identical. This local match suggests in turn hypothesizing that Event17 and Event5 match, and also that Event31 and Event63 match. Each suggested match leads to the creation of new local matches involving the arguments of the statement if either (a) both are entities (e.g., objects or constants), (b) both are terms involving functions, which are an indirect means of referring to entities or dimensions, or (c) both are expressions whose functors match. Here is an example of substitution involving functions:

```
B2: (PRESSURE Water32)
T2: (TEMPERATURE Brick45)
```

B2 and T2 could be placed into correspondence if they were the arguments of some other matching pair of statements since PRESSURE and TEMPERATURE are both functions (in this case referring to values on physical dimensions of the respective objects).

The idea that two statements can match only if their relational predicates are "sufficiently alike" is based on the claim that some common relational content is required in analogy. We disagree with Holyoak and Thagard's (1989) claim that pure structural isomorphisms can qualify as analogies. They have presented the following pair:

Bill is smart and tall.	Rover is hungry and friendly.
Steve is smart.	Fido is hungry.
Tom is timid and tall.	Blackie is frisky and friendly.

Holyoak and Thagard (1989, p. 343) noted that ACME (and five out of the eight subjects tested) could match this pair and agree on the best attribute correspondence. But the fact that it can be solved is not decisive: We would suggest that it is taken as a logical puzzle to be solved for the best correspondences, not as an analogy. The trouble with accepting pure graph matches is that it leads to the claim that pairs like (1) and (2) are analogies, which seems patently untrue:

(1) Fred loves New York. (2) General Motors sells cars.

Note that it is the *relational* meaning that must be shared; (2) and (3) form an analogy but (2) and (4) do not:

(3) Fred peddles popsicles. (4) General Motors heads the list.

The question is how to formalize this requirement of common relational content. Structure-mapping uses the idea of *tiered identicality.* The default criterion is "sufficiently alike" for predicates other than functions is that the predicates are identical. We call this the *simple identically* criterion. Simple identicality of conceptual relations is an excellent first-pass criterion because it is computationally cheap. The notion of simple identicality might suggest an inability to process any matches other than literal matches. This is not the case. First, we assume that input representations are canonical *conceptual* representations, not semi-verbal strings. Second, functions, which represent domain dimensions, can be matched nonidentically if they are embedded in matching relational structure. This ability to align nonidentical functions provides considerable flexibility. This is what allows SME to make cross-dimensional matches, as when we interpret "Sally is sharper than Bill" to mean that Sally is smarter than Bill. However, there are circumstances where criteria requiring more processing are worthwhile (e.g., when placing two items in correspondence would allow a larger, or very relevant, structure to be mapped, as in Falkenhainer's (1987, 1990a, b), work). In these circumstances weaker criteria (in that they allow more items to match) that involve more processing are allowed. One such test is *minimal ascension* (Falkenhainer, 1987, 1990a, b) which allows two items to be placed into correspondence if their predicates have close common superordinates. Another technique is *decomposition:* Two concepts that are similar but not identical (such as "bestow" and "bequeath") are decomposed into a canonical representation language so that their similarity is expressed as a partial identity (here, roughly, "give"). Decomposition is the simplest form of *re-representation* (Gentner, 1989; Gentner & Rattermann, 1991), where additional knowledge is used to reformulate a description in order to achieve a better match. In this article, we only use SME with the first-level identicality constraint. As Section 6 argues, this simple constraint seems to provide a better psychological account than more complex constraints do.

The process of using matches to propose lower matches is recursive, ending with entity matches. SME does not try matches between every pair of objects in base and target: It only hypothesizes object matches when there is some aspect of the relational structure that suggests that the objects might correspond. This leads to substantial efficiencies over purely bottom-up matchers, such as Winston's (analogy program, 1992).

The output of the first phase is a network of *match hypotheses,* each representing a local match between an item of the base and target. At this stage, the network is incoherent. The set of correspondences taken as a whole is structurally inconsistent, often including *N*-to-one mappings. Furthermore, this initial network may contain match hypotheses that are not *grounded* and so can never be part of any global interpretation. A match hypothesis is *grounded* if a recursive chain of correspondences from it through its arguments exists all the way down to entities. Only grounded match hypotheses can participate in global interpretations. Otherwise, global interpretations might include statements whose arguments did not match, which would violate the parallel connectivity constraint.

Looking at a simple example makes this process clearer. Figure 2 shows two drawings used in psychological experiments concerning analogy,[4] with a propositional representation of these pictures suitable for simulation shown in Figure 3. The right-hand side of Figure 3 shows the propositions in standard logical format, whereas the left-hand side contains an equivalent graphical representation which is useful for understanding the match process. Figure 4 illustrates the match hypotheses computed by SME for these descriptions.

Even though the initial network of match hypothesis is structurally inconsistent, it contains every consistent interpretation of the match; global interpretations emerge out of the initial network. Thus, the maximum size of any global interpretation, as measured in number of correspondences, is limited by the size of this network. We exploit this fact in Section 3.3.

In the second phase, these local matches are coalesced into global interpretations. The SME algorithm combines structurally consistent combinations of match hypotheses (i.e., sets with consistent object bindings and consistent relational argument assignments). For instance, in Figure 4 there are two match hypotheses involving Grant, one which places him in correspondence with Jack because PERSON is true of both of them, and another match hypothesis which places Grant in correspondence with RobotJ, because both are agents of the same kind of action, repairing. No interpretation of this comparision can include both of these match hypotheses. Merging can be done exhaustively, producing all possible interpretations (as in Faulkenhainer et al., 1986, 1989); however, we normally use a more psychologically plausible *greedy merge* algorithm, which produceds only one or two interpretations and operates in linear time (Forbus, Ferguson, & Gentner, 1994; Forbus & Oblinger, 1990).

[4] We thank Arthur Markman for the drawings of Figure 2 and the corresponding representations.

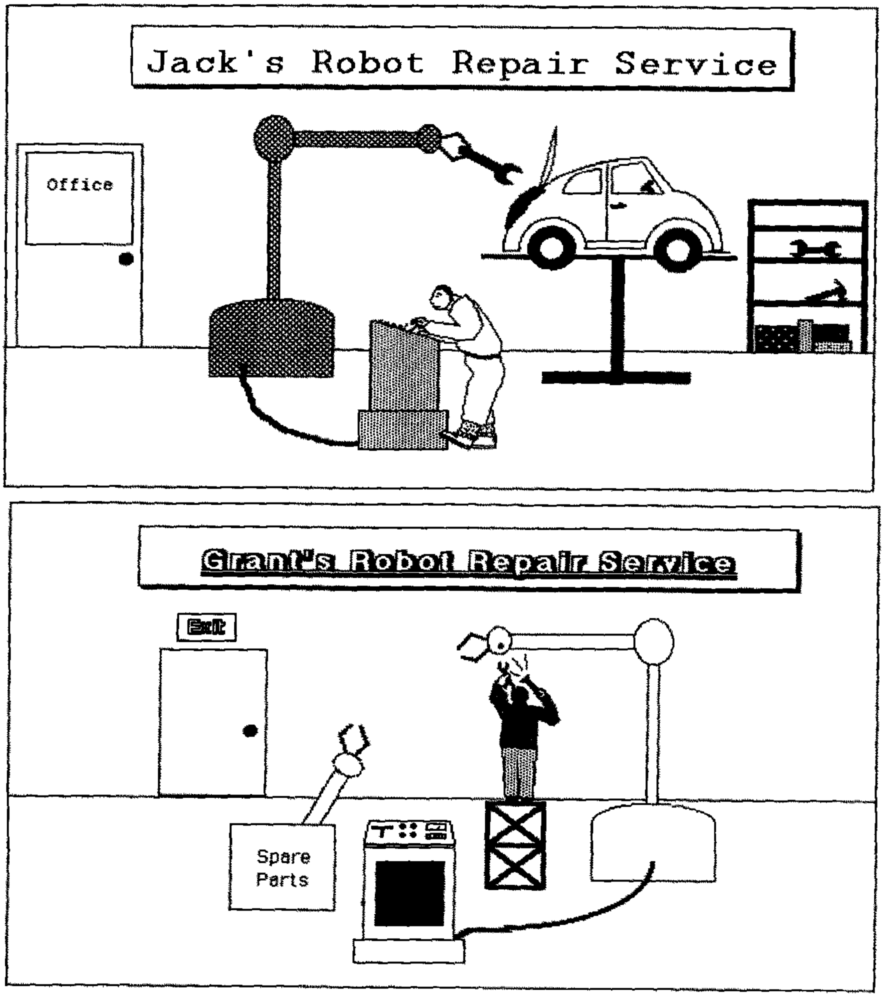
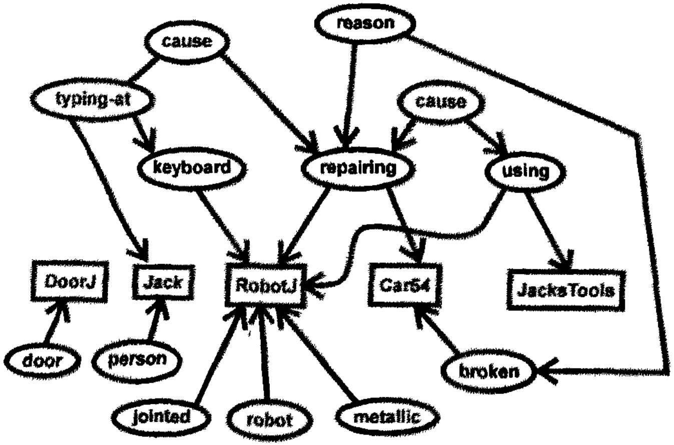


Figure 2. Two simple situations.

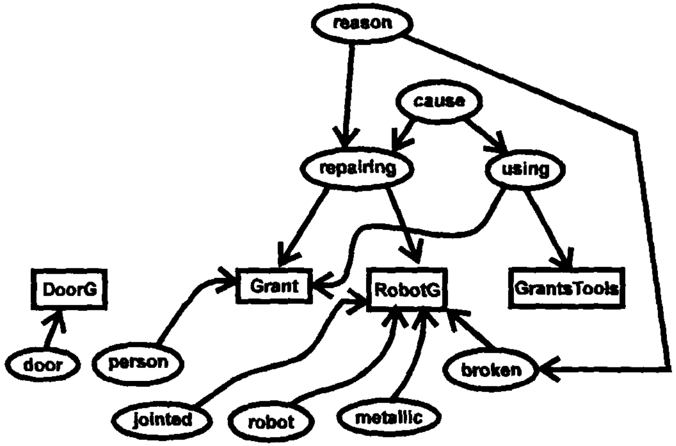
The third phase is structural evaluation. For simplicity, we describe this stage as conceptually distinct from the previous stage, although it is actually interleaved with building interpretations, because its results guide the greedy merge algorithm. To capture human preferences, the structural evaluation computation should favor interpretations with many matches over those with few matches and deep interpretations over shallow interpretations. The first step is to assign an initial score to every match hypothesis. This helps enforce the size preference. The systematicity preference is implemented via

Figure 3. Sample descriptions. Here are two predictable calculus descriptions given to SME to illustrate the algorithm's operation.



Jack's Robot Repair description:

```
(REASON (REPAIRING ROBOTJ CAR54)
        (BROKEN CAR54))
(CAUSE (TYPING-AT JACK
          (KEYBOARD ROBOTJ))
       (REPAIRING ROBOTJ CAR54))
(CAUSE (REPAIRING ROBOTJ CAR54)
       (USING ROBOTJ HANDTOOLSJ))
(DOOR DOORJ)
(JOINTED ROBOTJ)
(METALLIC ROBOTJ)
(ROBOT ROBOTJ)
(PERSON JACK)
```



Grant's Robot Repair description:

```
(REASON (REPAIRING GRANT ROBOTG)
        (BROKEN ROBOTG))
(CAUSE (REPAIRING GRANT ROBOTG)
       (USING GRANT HANDTOOLSG))
(TOOL-SET HANDTOOLSG)
(DOOR DOORG)
(JOINTED ROBOTG)
(METALLIC ROBOTG)
(ROBOT ROBOTG)
(PERSON GRANT)
```

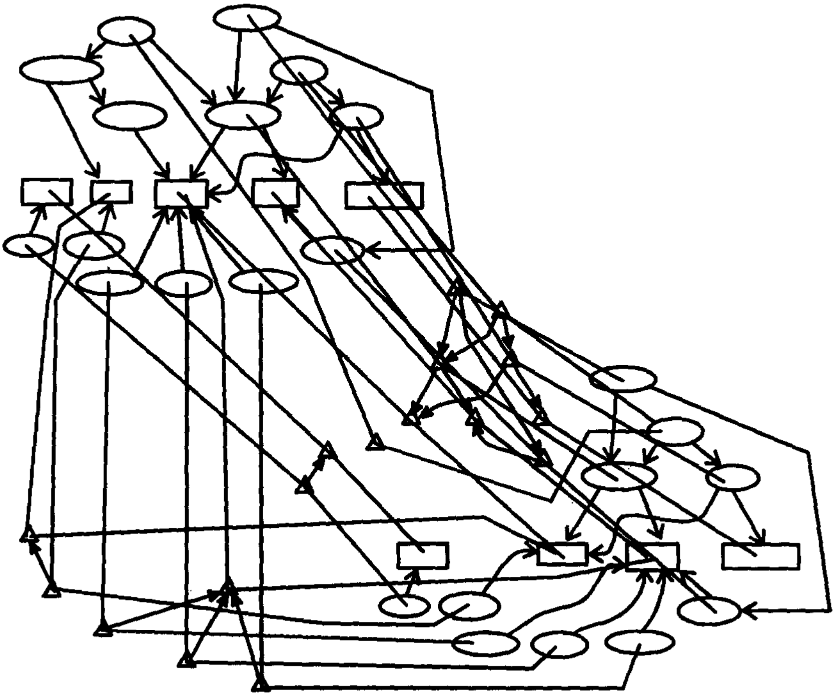


Figure 4. A match hypothesis forest. This picture illustrates the match hypotheses generated for a pair of simple descriptions. Match hypotheses are shown as triangles. Dashed lines indicate the base and target items each match hypothesis places in correspondence. The solid arrows leaving a match hypothesis indicates what others it relies upon to be structurally consistent. Notice that the one of the match hypotheses involving the occurrence of CAUSE in the target is structurally inconsistent, because its arguments cannot be aligned.

a *trickle-down* method: Match hypothesis scores are passed down to increment the scores of matching arguments.[5] That is, if $W(MH_1)$ is the score associated with a match hypothesis MH_1, MH_2 is a match hypothesis that applies to one of MH_1's arguments, and δ is the trickle-down factor, then $W(MH_2)$ is incremented as follows:

$$W(MH_2) \leftarrow \max\{W(MH_2) + \delta W(MH_1);\ 1.0\}$$

This local computation causes scores to cascade downwards, providing higher values to those object correspondences which support the alignment

[5] The systematicity preference could have been implemented by differentially weighting matches at different levels. This method would seem to require a computationally implausible "bird's-eye" view of the representations. In a comparison of the two methods, the trickle-down method accounted for human soundness ratings better than treating weights directly as a function of order (Forbus & Gentner, 1989).

```
GM1: 10 correspondences, SES = 4.66
  Object mappings:
   DOORG <-> DOORJ
   ROBOTG <-> CAR54
   GRANT <-> ROBOTJ
   HANDTOOLSG <-> HANDTOOLSJ
  No candidate inferences.
```

Figure 5. Global interpretations for the example. Here is a summary of the best interpretations for this match found by SME. SES refers to the structural evaluation of the interpretation.

of large relational structures. The structural evaluation of a global interpretation is simply the sum of the scores of the match hypotheses which comprise its correspondences.

The final phase is the computation of candidate inferences. Computing candidate inferences requires knowing the set of correspondences, so this takes place after the merge operation. Candidate inferences are generated by finding noncorresponding relational structure in the base which can be conjectured to hold the target. The global interpretations built for the comparison of Figure 3 are shown in Figure 5. In this simple example, there are no candidate inferences.

It is important to note that the literal similarity computation can produce purely relational interpretations as well as overal similarity interpretations, and that it can produce purely surface interpretations as well. It is simply a question of which collection of local matches wins. This reflects the human ability to process a novel comparison and discover only after the fact that it is an analogy. We assume that this all-purpose literal similarity mode is the normal mode of similarity processing in the absence of specific instructions. Consequently, SME creates initial local matches for attribute statements as well as for relational statements.

For SME to play a major role in a model of similarity-based retrieval, it should be consistent with psychological evidence. We have tested the psychological validity of SME as a simulation of analogical processing in several ways. For instance, we compared SME's structural evaluation scores with human soundness ratings for the "Karla the Hawk" stories discussed later (Gentner & Landers, 1985; Rattermann & Gentner, 1987). Like humans, SME rated analogical matches higher than surface matches (Skorstad, Falkenhainer, & Gentner, 1987). The patterns of preference were similar across story sets: There was a significant positive correlation between the difference scores for SME and those for human subjects, where the difference score is the rating for analogy minus the rating for surface match within a given story set (Gentner, Rattermann, & Forbus, 1993).

Because retrievals occur frequently, components in model of retrieval must be efficient. SME is quite efficient. The generation of match hypotheses is $O(n^2)$ on a serial machine, where n is the number of items in base or target and should typically be better than O(log(n)) on data-parallel machines.[6] The generation of global interpretations is roughly O(log(n)) on a serial machine, using the greedy merge algorithm of Forbus and Oblinger (1990),[7] and even faster parallel merge algorithms seem feasible.

3.2 The FAC Stage

The FAC stage is essentially a bank of SME matchers, all running in parallel in literal similarity mode.[8] These take as input the memory descriptions that are passed forward by the MAC stage and compute a structural alignment between each of these descriptions and the probe. The other component of the FAC stage is a *selector*—currently a numerical threshold—which chooses some subset of these comparisons to be available as the output of the retrieval system (see Figure 1).

The FAC stage acts as a structural filter. It captures the human sensitivity to structural alignment and inferential potential (subject to the limited and possibly surface-heavy set of candidates provided by the MAC stage, as described later). Several remarks on this algorithm's role in retrieval are in order. We use the literal similarity algorithm, on the grounds that in reminding situations people can respond to and identify different kinds of similarity. (Recall that the literal similarity computation can compute relational similarity or object similarity as well as overall similarity). This choice seems ecologically sound because mundane matches are often reasonable guides to action; riding a new bicycle, for instance, is like riding other bicycles (Forbus & Gentner, 1986; Gentner, 1989; Medin & Ortony, 1989; Medin & Ross, 1989). Finally, this choice is necessary to model the high observed frequency of surface remindings. These surface remindings would mostly be rejected if FAC were strictly an analogy matcher. The selector for the FAC stage must choose a small set of matches for subsequent processing. Currently we select as output the best match, based on its structural evaluation, and any others within 10% of it. We settled on the 10% criteria because it generally returns a single result, only producing multiple results when there are two extremely close candidates. However, other criteria are possible, and we have experimented with broadening the percentage, selecting a fixed number, selecting a maximum number (if capacity limits were assumed), and so forth. (One class of these experiments is described in Section 5.) We have also considered adding a threshold to the selector, so that if the best outcome is too weak, the retrieval system returns nothing.

[6] The worst-case parallel time would be O(n), in degenerate cases where all but one of the local matches is proposed by matching arguments.

[7] The original exhaustive merge algorithm was worst-case factorial in the number of "clumps" of match hypotheses but, in practice was often quite efficient. See Falkenhainer et al. (1989) for details.

[8] In our current implementation, SME is run sequentially on each candidate item in turn, but this is an artifact of the implementation.

3.3 The MAC Stage

The MAC stage collects the initial set of matches between the probe and memory. Like the FAC stage, the MAC stage conceptually consists of a set of matchers and a selector that simply returns all items whose MAC score is within 10% of the best score given that probe. The challenge of the MAC stage is in the design of its matcher. It must allow quickly comparing, in parallel, the probe to a large pool of descriptions and passing only a few on to the more expensive FAC stage. The rest of this section describes the design and implementation of the MAC matcher.

Let us start by examining in more detail the design criteria the MAC matcher must satisfy. Ideally, we would like the most similar or apt memory item for the given probe. Clearly, running SME on the probe and every item in memory would prove the most accurate result. Unfortunately, even though SME is very efficient, it isn't efficient enough. SME operates by building intermediate structure, in the form of the network of local matches. The idea of building such networks for a pair of items, or a small number of pairs of items, is psychologically plausible, because the size of the match hypothesis network is polynomial in the size of the descriptions being matched. This means, depending on one's implementation assumptions, that a fixed-size piece of hardware could be built which could be dynamically reconfigured to represent any local match network for input descriptions of some bounded size. What is not plausible is that such networks could be built between a probe and every item in a large memory pool, and especially that this could happen quickly enough in neural architectures to account for observed retrieval times (cf. Minsky, 1981; Waltz, 1989).

This architectural argument suggests that, while SME in literal similarity mode is fine for FAC, MAC must be made of simpler stuff. To escape having to suffer the complexity of the most accurate matcher in the "innermost loop" of retrieval, we must trade accuracy for efficiency. The MAC matcher must provide a crude, computationally cheap match process to pare down the vast set of memory items into a small set of candidates for more expensive processing. Ideally, MAC's computations should be simple enough to admit plausible parallel and/or connectionist implementations for large-scale memory pools.

What is the appropriate crude estimator of similarity? The most straightforward method would be to count the number of match hypotheses that FAC would generate in comparing a probe to a memory item. Let us call this number the *numerosity* of a comparison. Numerosity bears a rough relation to the potential size of the global interpretation, because the more local matches there are, the larger the global interpretation could potentially be. However, a large number of match hypotheses does not guarantee a large global interpretation, for two reasons. First, many match hypotheses might be ungrounded (recall Section 3.1) and hence cannot be part of any global interpretation. Second, often many combinations of match hypotheses are ruled out by the 1:1 constraint, working against the formation of large global interpretations. Both reasons follow directly from the fact that numerosity is not structurally sensitive. However, something like numerosity is at least a crude estimate of similarity.

One straightforward way to implement a rough similarity estimator would be to calculate numerosity by building the actual match hypothesis network (e.g., to carry out the first part of a full analogy process) for the probe and each memory item and then count the match hypotheses. This is what our original version of MAC/FAC did (Gentner, 1989a). It also is roughly what ARCS (Thagard, Holyoak, Nelson, & Gochfeld, 1990) does. ARCS models retrieval by building a network of connections similar to SME's match hypothesis network between the probe and each item in the memory pool that shares a semantically similar predicate with it.[9] As just discussed, we view these solutions as psychologically and computationally implausible. Even with parallel and/or neural hardware, it is hard to see how to generate match hypothesis networks between a probe and everything in a large pool of memory, while still providing realistic response times. A cheaper method is required.

We have developed a novel technique for estimating the degree of match in which structured representations are encoded as *content vectors*. Content vectors are flat summaries of the knowledge contained in complex relational structures. The content vector for a given description specifies which functors (i.e., relations, connectives, object attributes, functions, etc.) were used in that description and the number of times they occurred. Content vectors are assumed to arise automatically from structured representations and to remain associated with them. Content vectors are a special form of feature vectors.

More precisely, let Π be the set of functors used in the descriptions that constitute memory items and probes. We define the *content vector* of a structured description as follows. A content vector is an n-tuple of numbers, each component corresponding to a particular element of Π. Given a description ϕ, the value of each component of its content vector indicates how many times the corresponding element of Π occurs in ϕ. Components corresponding to elements of Π which do not appear in statements of ϕ have the value zero. One simple algorithm for computing content vectors is to count the number of occurrences of each functor in the description. Thus, if there were four occurrences of IMPIES in a story, the value for the IMPLIES component of its content vector would be 4. (Figure 6 illustrates.) Thus, content vectors are easy to compute from a structured representation and can be stored economically (using sparse encoding, for instance, on serial machines).

[9] ARCS is based on Holyoak and Thagard's (1989) ACME, an analogy matcher which uses a localist connectionist network similar to SME's match hypothesis network to construct a single interpretation of a comparison via constraint satisfaction.

Solar System: Structured representation

```
(CAUSE
  (GRAVITY (MASS SUN) (MASS PLANET))
  (ATTRACTS SUN PLANET))
(GREATER (TEMPERATURE SUN)
         (TEMPERATURE PLANET))
(CAUSE (AND (GREATER (MASS SUN)
                     (MASS PLANET))
            (ATTRACTS SUN PLANET))
       (REVOLVE-AROUND PLANET SUN))
```

Solar System: Content Vector

```
(AND . 1)
(ATTRACTS . 1)
(CAUSE . 2)
(GRAVITY . 1)
(GREATER . 2)
(MASS . 2)
(OBJECTS . 2)
(REVOLVE-AROUND . 1)
(TEMPERATURE . 2)
```

Rutherford Atom: Structured representation

```
(CAUSE (OPPOSITE-SIGN (CHARGE NUCLEUS)
                      (CHARGE ELECTRON))
       (ATTRACTS NUCLEUS ELECTRON))
(REVOLVE-AROUND ELECTRON
                NUCLEUS)
(GREATER (MASS NUCLEUS)
         (MASS ELECTRON))
```

Rutherford Atom: Content Vector

```
(ATTRACTS . 1)
(CAUSE . 1)
(CHARGE . 2)
(GREATER . 1)
(MASS . 2)
(OBJECTS . 2)
(OPPOSITE-SIGN . 1)
(REVOLVE-AROUND . 1)
```

Figure 6. Sample representations with content vectors. Here are some simple predicate calculus representations and the corresponding content vectors. A simple counting algorithm is used here, in the simulation these are normalized to unit vectors.

How good an approximation is the content vector dot product to what SME would produce? Suppose content vectors were generated using the simple counting algorithm described above. Then the product of each corresponding component is an overestimate of the number of match hypotheses that would be created between functors of that type, because it does not take into account the cases when the arguments to the match hypotheses could not be aligned. There is also a possibility of underestimation, because the dot product does not take into account matches between nonidentical functions and entities, because discovering those matches requires tracing predicate bindings. However, in practice, the number of entity and nonidentical function matches tends to be smaller than the number of ungrounded matches, so overall, the dot product tends to overestimate numerosity and hence will tend to be an overestimate of what SME would produce.

The dot product of content vectors provides exactly the computational basis the MAC stage needs. It could be implemented efficiently for large memories using a variety of massively parallel computation schemes. For instance, connectionist memories can be built which find the closest feature vector to a probe (Hinton & Anderson, 1989). Therefore, the MAC stage can scale up.

To summarize, the MAC matcher works as follows: Each memory item has a content vector stored with it.[10] When a probe enters, its content vector

[10] We normalize content vectors to unit vectors, both to reduce the sensitivity to overall size of the descriptions and because we assume that psychologically plausible implementation substrate for MAC/FAC (e.g., neural systems) will involve processing units of limited dynamic range.

TABLE 1
Types of Stories Used in the "Karla the Hawk" Experiments

	Common First-Order Relations	Common Higher-Order Relations	Common Object Attributes
LS	Yes	Yes	Yes
SF	Yes	No	Yes
AN	Yes	Yes	No
FOR	Yes	No	No

Note. LS=literal similarity; SF=surface similarity; AN=analogy; FOR=first-order relations.

is computed. A score is computed for each item in the memory pool by taking the dot product of its content vector with the probe's content vector. The MAC selector then produces as output the best match and everything within 10% of it, as described previously. (As for the FAC stage, variants that could be considered include adding a bound on the number of items returned (to model capacity limitations) and implementing a threshold on the MAC selector so that if every match is too low MAC returns nothing.)

Like other feature-vector schemes, the dot product of content vectors does not take the actual relational structure into account. It only calculates a numerical score and hence doesn't produce the correspondences and candidate inferences that provide the power of analogical reasoning and learning. But the output of MAC feeds to the FAC stage, which operates on structured representations. Thus, it is the FAC stage that both filters out structurally unsound remindings and produces the desired correspondences and candidate inferences. We claim that the interplay of the cheap but dumb computations of the MAC stage and the more expensive but structurally sensitive computations of the FAC stage explains the psychological phenomena of Section 2. As the first step in supporting this claim, we next demonstrate that MAC/FAC's behavior provides a good approximation of psychological data.

4. COGNITIVE SIMULATION EXPERIMENTS

In this section, we compare the performance of MAC/FAC with that of humans, using the "Karla the Hawk" stories (Gentner, Rattermann, & Forbus, 1993; Rattermann & Gentner, 1987, Experiment 2). For these studies, we wrote sets of stories consisting of base stories plus four variants, created by systematically varying the kind of commonalities. All stories shared first-order relations (primarily events) but varied in which other commonalities were present, as shown in Table 1. The LS (literal similarity) stories shared both higher-order relational structure and object attributes. The AN (analogy) stories shared higher-order relational structure but contained different attributes, whereas the SF (surface similarity) stories shared

TABLE 2
Proportion of Remindings for Different Match Types: Human Participants

Condition	Proportion
LS	.56
SF	.53
AN	.12
FOR	.09

Note. LS=literal similarity; SF=surface similarity; AN=analogy; FOR=first-order relations.

attributes but contained different higher-order relational structure. The FOR (first-order relations) stories differed both in attributes and higher-order relational structure.

In this study, the subjects were first given 32 stories to remember, of which 20 were base stories and 12 were distractors. They were later presented with 20 probe stories which matched the base stories as follows: 5 LS matches, 5 AN matches, 5 SF matches, and 5 FOR matches. They were told to write down any prior stories of which they were reminded. (Which stories were in each similarity condition was varied across subjects.) As shown in Table 2, the proportions of remindings for different match types were .56 for LS, .53 for SF, .12 for AN, and .09 for FOR. Table 2 also shows that this retrievability order has been stable across three variations of this study: LS ≥ SF > AN ≥ FOR.[11]

As discussed above, this retrievability order differs strikingly from the soundness ordering. When subjects were asked to rate how *sound* the matches were—how well the inferences from one story would apply to the other—they rated analogy (AN) and literal similarity (LS) as significantly more sound than surface similarity (SF) and FOR matches (matches based only on common first-order relations, primarily events). SME running in analogy mode on SF and AN matches correctly reflected human soundness rankings (Forbus & Gentner, 1989; Gentner et al., in press; Skorstad et al., 1988). Here we seek to capture human retrieval patterns: Does MAC/FAC duplicate the human propensity for retrieving SF and LS matches rather than AN and FOR matches? The idea is to give MAC/FAC a memory set of stories, then probe with various new stories. To count as a retrieval, a story must make it through both MAC and FAC. We use replication of the ordering found in the psychological data, rather than the exact percentages, as our criterion for success because this measure is more robust, being less sensitive to the detailed properties of the databases.

[11] LS and SF did not differ significantly in retrievability. In Experiment 2, AN and FOR did not differ significantly, although in Experiment 1, AN matches were better retrieved than FOR matches.

```
(FOLLOW
  (PROMISE MAN1 KARLA
    (NOT (ATTACK MAN1 KARLA)))
  (ATTACK MAN1 DEER))
(CAUSE
 (EQUALS (HAPPINESS MAN1) HIGH)
 (PROMISE MAN1 KARLA
   (NOT (ATTACK MAN1 KARLA))))
(CAUSE
 (OBTAIN MAN1 FEATHERS)
 (EQUALS (HAPPINESS MAN1) HIGH))
(FOLLOW
 (OFFER KARLA FEATHERS MAN1)
 (OBTAIN MAN1 FEATHERS))
(CAUSE
 (REALIZE KARLA
     (DESIRE MAN1 FEATHERS))
 (OFFER KARLA FEATHERS MAN1))
(FOLLOW
 (EQUALS
  (SUCCESS
     (ATTACK MAN1 KARLA)) F)
 (REALIZE KARLA
   (DESIRE MAN1 FEATHERS)))
(CAUSE
 (NOT (USED-FOR
         FEATHERS CROSS-BOW))
 (EQUALS (SUCCESS
           ATTACK MAN1 KARLA))
               F))
(FOLLOW
 (ATTACK MAN1 KARLA)
 (EQUALS (SUCCESS
     (ATTACK MAN1 KARLA))
               F))
(FOLLOW
  (SEE KARLA MAN1)
  (ATTACK MAN1 KARLA))
(HAPPEN (SEE KARLA MAN1))
(LIVES KARLA LOC1)
(POSSESS MAN1 CROSS-BOW)
(POSSESS KARLA FEATHERS)
(RUMINANT DEER)
(ANTLERED DEER)
(HOOFED DEER)
(QUADRIPED DEER)
(MAMMAL DEER)
(THIN CROSS-BOW)
(LARGE CROSS-BOW)
(MEDIEVAL CROSS-BOW)
(WOODEN CROSS-BOW)
(WEAPON CROSS-BOW)
(BLACK FEATHERS)
(COVERING FEATHERS)
(LONG FEATHERS)
(SOFT FEATHERS)
(ASSET FEATHERS)
(VOCAL MAN1)
(BIPED MAN1)
(HUNTER MAN1)
(WARLIKE MAN1)
(HUMAN MAN1)
(MALE MAN1)
(PREDATORY KARLA)
(BLACK KARLA)
(POWERFUL KARLA)
(LARGE KARLA)
(HAWK KARLA)
```

Figure 7. A representation from the Karla the Hawk story set.

For the computational experiments, we encoded predicate calculus representations for 9 of the 20 story sets (45 stories). Figure 7 shows one of the story representations. These stories are used in all three of the following experiments.

4.1 Cognitive Simulation Experiment 1

In our first study, we put the nine basic stories in memory, along with the nine FOR stories which served as distractors. We then used each of the variants—LS, SF, and AN—as probes. This roughly resembles the original task, but MAC/FAC's job is easier in that (a) it has only 18 stories in memory, whereas participants had 32, in addition to their vast background knowledge; and (b) participants were tested after a week's delay, so that there could have been some degradation of the memory representations.

Table 3 shows the proportion of times the base story made it through the MAC and (then) through FAC. The FAC output is what corresponds to

TABLE 3
Proportion of Correct Retrievals
Given Different Kinds of Probes

Probes	MAC	FAC
LS	1.0	1.0
SF	0.89	0.89
AN	0.67	0.67

Note. LS=literal similarity; SF=surface similarity; AN=analogy; FOR=first-order relations. Memory contains 9 base stories and 9 FOR matches; probes were the 9 LS, 9 SF, and 9 AN stories. The rows show proportion of times the correct base story was retrieved for different probe types.

TABLE 4
Mean Numbers of Different Match Types Retrieved
Per Probe When Base Stories are Used as Probes

Retrievals	MAC	FAC
LS	0.78	0.78
SF	0.78	0.44
TA	0.33	0.22
FOR	0.22	0.0
Other	1.33	0.22

Memory contains 36 base stories (LS, SF, AN, and FOR for 9 story sets); the 9 base stories used as probes. Other=any retrieval from a story set different from the one to which the base belongs.

human retrievals. MAC/FAC's performance is much better than that of the human participants, perhaps partly because of the differences noted above. However, the key point is that its results show the same ordering as those of humans: LS > SF > AN.

4.2 Cognitive Simulation Experiment 2

To give MAC/FAC a harder challenge, we put the four variants of each base story into memory. This made a larger memory set (36 stories) and also one with many competing similar choices. Each base story in turn was used as a probe. This is almost the reverse of the task participants faced and is more difficult.

Table 4 shows the mean number of matches of different similarity types that succeed in getting through MAC and (then) through FAC. There are several interesting points to note here. First, the retrieval results (i.e., the number that make it through both stages) ordinally match the results for human participants: LS > SF > AN > FOR. This degree of fit is encouraging, given the difference in task. Second, as expected, MAC produces

TABLE 5
Mean Numbers of Different Match Types Retrieved Per Probe With Base Stories as Probes and No LS Stories in Memory

Retrievals	MAC	FAC
SF	0.88	0.78
AN	0.56	0.56
FOR	0.22	0.11
Other	1.11	0.11

Memory contains 27 stories (9 SF, 9 AN, 9 FOR); 9 base stores used as probes.

some matches that are rejected by FAC. This number depends partly on the criteria for the two stages. Here, with MAC and FAC both set at 10%, the mean number of memory items produced by MAC is 3.4, and the mean number accepted by FAC is 1.6. Third, as expected, FAC succeeds in acting as a structural filter on the MAC matches. It accepts all of the LS matches MAC proposes and some of the partial matches (i.e., SF and AN), while rejecting most of the inappropriate matches (i.e., FOR and matches with stories from other sets).

4.3 Cognitive Simulation Experiment 3

In the prior simulations, LS matches were the resounding winner. Although this is reassuring, it is also interesting to know which matches would be retrieved if there were no perfect overall matches. Therefore, we removed the LS variants from memory and repeated the second simulation experiment, again probing with the base stories. AS Table 5 shows, SF matches are now the clear winners in both the MAC and FAC stages. Again, the ordinal results match well with those of subjects: SF > AN > FOR.

4.4 Summary of Cognitive Simulation Experiments

The results are encouraging. First, MAC/FAC's retrieval results (i.e., the number that make it through both stages) ordinally match the results for human subjects: LS > SF > AN > FOR. Second, as expected, MAC produces some matches that are rejected by FAC. The mean number of memory items produced by MAC is 3.4, and the mean number accepted by FAC is 1.6. Third, FAC succeeds in its job as a structural filter on the MAC matches. It accepts all of the LS matches proposed by MAC and some of the partial matches (the SF, AN, and FOR matches) and rejects most of the inappropriate matches (the "other" matches from different story sets). It might seem puzzling that FAC accepts more SF matches than AN matches, when it normally would prefer AN over SF. The reason is that it is not generally being offered this choice. Rather, it must choose the best from thematches passed on by MAC for a given probe (which might be AN and LS, or SF and LS, for example).

It is useful to compare MAC/FAC's performance with that of Thagard et al.'s (1990) ARCS model of similarity-based retrieval, the most comparable alternate model. Thagard et al. gave ARCS the "Karla the Hawk" story in memory along with 100 fables as distractors. When given the four similarity variants as probes, ARCS produced asymptotic activations as follows: LS (.67), FOR (−.11), SF (−.17), AN (−.27). ARCS thus exhibits at least two violations of the LS $\geq$ SF $>$ AN $\geq$ FOR order found for human remindings. First, SF remindings, which should be about as likely as LS remindings, are quite infrequent in ARCS—less frequent than even the FOR matches. Second, AN matches are less frequent than FOR matches in ARCS, whereas for humans, AN was always ordinally greater than FOR and (in Experiment 1) significantly so. Thus, MAC/FAC explains the data better than ARCS. This is especially interesting because Thagard et al. argued that a complex localist connectionist network which integrates semantic, structural, and pragmatic constraints is required to model similarity-based remindings. Although such models are intriguing, MAC/FAC shows that a simpler model can provide a better account of the data. We compare MAC/FAC with ARCS in more detail in Section 6.

Finally, and most importantly, MAC/FAC's overall pattern of behavior captures the motivating phenomena. It allows for structured representations and for processes of structural alignment and mapping over these representations, thus satisfying the *structural representation* and *structured mappings* criteria. It produces fewer analogical matches than literal similarity or surface matches, thus satifying the *existence of rare insights* criterion. The majority of its retrievals are LS matches, thus satisfying the *primacy of the mundane* criterion. It also produces a fairly large number of SF matches, thus satisfying the *surface superiority* criterion. Finally, its algorithms are simple enough to apply over large-scale memories, thus satisfying the *scalability* criterion.

5. SENSITIVITY ANALYSES

The experiments of the previous section show that the MAC/FAC model can account for psychological retrieval data. This section looks more closely into *why* it does, by seeing how sensitive the results are to different factors in the model. These analyses are similar in spirit to those carried out by Van Lehn (1989) in his SIERRA project. Van Lehn used his model to generate different possible learning sequences to see if these variations covered the space of observed mistakes made by human learners in subtraction problems. Thus, variations in the model were used to generate hypotheses about the space of individual differences. Our methodology is quite similar, in that we vary aspects of our model in order to better understand how it accounts for data. The key difference is that we are not attempting to model individual differences but instead are investigating how our results depend on different aspects of the theory. Such *sensitivity analyses* are routinely used in other areas of science and engineering; we believe they are also an important tool for cognitive modeling.

Sensitivity analyses can provide insight into why a simulation works. Any working cognitive simulation rests on a large number of design choices. Examples of design choices include the setting of parameters, the kinds of data provided as input, and even the particular algorithms used. Some of these design choices are forced by the theory being tested, some choices are only weakly constrained by the theory, and others are irrelevant to the theory being tested but are necessary to create a working artifact. Sensitivity analyses can help verify that the source of a simulation's performance rests with the theoretically important design choices. Varying theoretically forced choices should lead to a degradation of the simulation's ability to replicate human performance. Otherwise, the source of the performance lies elsewhere. On the other hand, varying theoretically irrelevant choices should not affect the results, and if it does, it suggests that something other than the motivating theory is responsible for the simulator's performance. Finally, seeing how the ability to match human performance varies with parameters that are only weakly constrained by theory can lead to insights about why the model works.

In the rest of this section, we describe a series of sensitivity experiments on MAC/FAC. These experiments demonstrate that its ability to replicate human performance is robust, and that this ability depends crucially on the theoretically important design choices. We first describe the methodology used in these experiments in detail and then describe three sensitivity analyses.

5.1 Method for Sensitivity Analyses

A sensitivity analysis requires a standard of comparision, a baseline against which to judge the results of variations. We use as our baseline the simulation experiments described in Section 4. We say that a particular set of design choices *satisfies the data* if re-running the simulation experiments with that set of design choices yields results that match the human data. That is, the frequency of retrievals must follow the pattern LS > SF > AN > FOR.

There are many design choices which could be explored via sensitivity analyses. Conceptually, one can think of sets of design choices as points in a high dimensional space. In essence, the simulation studies of Section 4 provide information about one point in the design space. This metaphor is excellent for choices of numerical parameters, because these dimensions can be viewed as continuous. This metaphor is not as useful for other kinds of design choices, for example, algorithmic choices, because systematically enumerating the set of plausible algorithms for a task is quite difficult. To best visualize the results, choosing two numerical dimensions to vary allows patterns of satisfaction to be displayed as a table, whose entries represent measurements of the ability of the model to satisfy the data at sampled points in the design space.

The two most interesting numerical parameters with respect to sensitivity analyses on MAC/FAC are the selector widths for the MAC and FAC stages, because these are only weakly constrained by the theory. They should be narrow, in order to reject inappropriate remindings, but we currently see no theoretically motivated method to calculate precise predictions for these parameters. Therefore, in these analyses we use an empirical approach. We vary the selector widths, using these variations as the axes of a subset of the design space. Recall that a selector of width W accepts all matches within W% of the largest input. That is, a selector with width 10% outputs the best match plus any other matches that are within 10% of the score of the best match, while a selector of width 100% will simply pass through all of its inputs. In the experiments below, selector widths for both MAC and FAC are varied from 1 to 100%, in 10% increments. Each entry in the table indicates whether that pair of width settings, combined with the other design choices, satisfied the data. When the pattern of retrieval is violated, the table entry contains information about the particular kind of violation.

Viewed as a map, the table of results from the sensitivity analysis can be divided into *viable regions*, subspaces of design choices which allow the model to satisfy the data, and *nonviable regions*, where they do not. The existence of viable regions is of course critical for a successful simulation. However, the nature of the nonviable regions is also interesting, because they provide a source of insight into why the model works. Seeing how a bridge collapses after replacing a particular strut with a weaker material (preferably via simulation) supports the conclusion that the strength of that strut was a factor in preventing collapse.

It should be noted that the computational costs of these experiments is large but not horrendous. Essentially, the cognitive simulation experiments of Sections 4.1 and 4.2 were replicated for each pair of selector widths, that is, 121 times. Each repetition required running the MAC matcher 810 times,[12] for a total of 98,010 times. The number of FAC executions varies with the size of the set output from MAC, of course, and varies substantially according to the particular design choices made (as shown later). A reasonably accurate estimate for the lower bound of FAC executions for each experiment is 900, and a reasonable upper bound is 1,600. The MAC matcher takes roughly 0.002 s for each pair of content vectors, and the FAC matcher (i.e., SME) takes between 1.0 and 11 s for each pair of structured representations, with an average time of roughly 4 s.[13] Thus, the time to run MAC/FAC for each probe typically ranges from 3 to 10 s. A naive system for doing sensitivity

[12] The first experiment involves 486 MAC executions because there are 18 stories in memory and 27 probes. The second experiment involves 324 MAC executions because there are 36 stories in memory and 9 probes.

[13] These times are for an IBM RS/6000 Model 350, SME3b, which was used in all experiments in this section. An earlier version of SME was used in Forbus and Gentner (1989) and in the experiments in Section 6.

Rows: MAC widths. Columns: FAC widths.

	1%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1%	4	4	4	4	4	4	4	4	4	4	4
10%	4	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
20%	4	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
30%	4	64	64	80	80	80	112	112	112	112	112
40%	4	64	64	80	80	80	112	112	368	368	368
50%	4	64	64	80	80	80	112	368	368	368	368
60%	4	64	64	80	80	80	112	368	368	368	368
70%	4	64	64	80	80	80	112	368	368	368	368
80%	4	64	64	80	80	80	112	368	368	368	368
90%	4	64	64	80	80	80	112	368	368	368	368
100%	4	64	64	80	80	80	112	368	368	368	368

Legend
Y = Satisfies the data
4 = No analogies
64 = SF ≤ AN
80 = LS ≤ AN , SF ≤ AN
112 = LS ≤ AN , SF ≤ AN , LS ≤ FOR
368 = LS ≤ AN , SF ≤ AN , LS ≤ FOR , LS ≤ DT

Rows are the width of the MAC selector, Columns correspond to the width of the FAC selector. The codes describe whether that combination of selector widths allows MAC/FAC to account for the human data, and if not, what criteria were violated.

Table 6. Sensitivity to selector width, normalizerd content vectors.

analyses could use as much as 5 h per analysis (14,000 s for MAC, 4,000 s for FAC). However, we found that by caching the results of matches in a simple database, we could cut the CPU requirements for these analyses considerably.

5.2 Sensitivity Analysis One: Robustness

In this experiment, we tested the robustness of MAC/FAC's ability to satisfy the data by varying the selector widths. Table 6 shows the results. Notice that there is one region that satisfies the data: When the MAC width is between 10% and 20% and FAC is at least 10%. The moderately large viable subspace indicates that MAC/FAC's performance is robust and not hostage to a particular choice of selector width settings.

As discussed previously, it is important to show that there are parameter settings that do not fit the human data, to establish that the theoretical variables actually matter. When either MAC or FAC is too narrow (i.e., MAC of 1% or FAC of 1%), analogies are never retrieved. This violates the rare insights criterion. When MAC is broad (30% or larger), making FAC too broad leads first to too many analogies, and then to junk remindings. The shape of the region of viability suggests that although FAC is necessary to provide structural matching and candidate inferences, MAC provides most of the filtering. Because that is MAC's intended purpose, this provides further evidence that the simulation works according to the principles of its design, rather than some unknown factor.

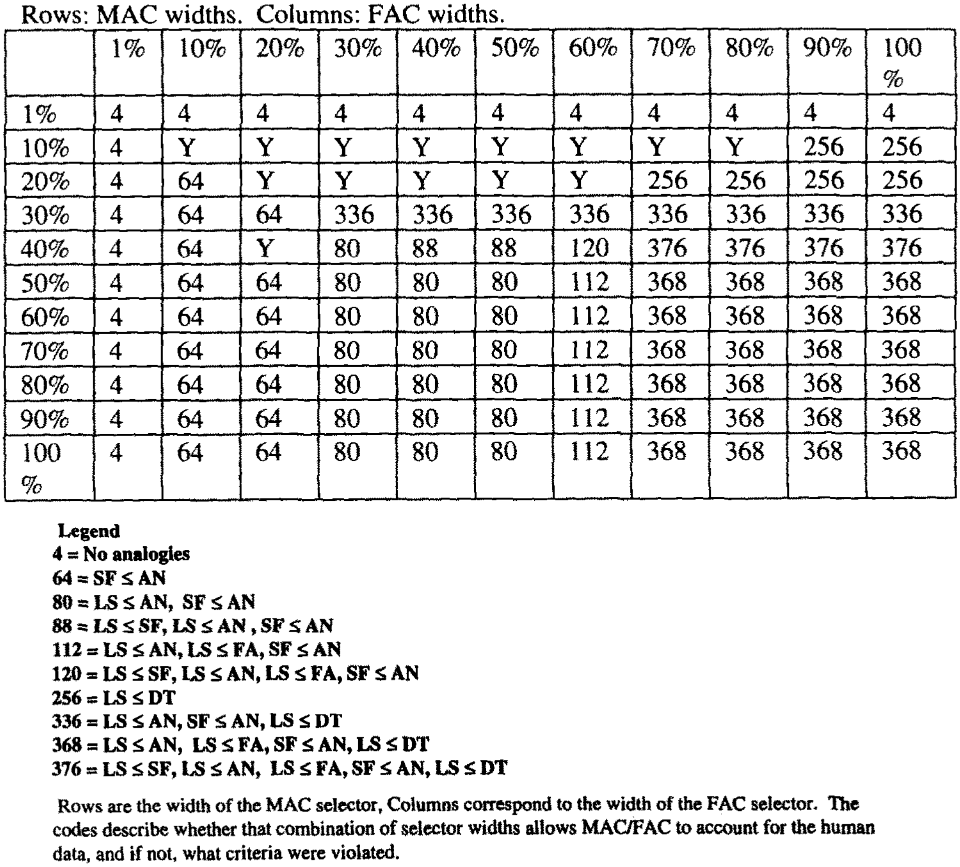
The evidence that the results are not very sensitive to the particular choice of selector widths in the original experiments (i.e., 10% for both MAC and FAC) is reassuring. The next two sensitivity analyses explore other design choices, using the same methodology as this experiment.

5.3 Sensitivity Analysis Two: Irrelevance of Normalization Details

In other sensitivity experiments on analogical processing algorithms (Forbus & Gentner, 1989), we demonstrated that the choice of normalization algorithm could affect outcomes in simulations of structural evaluation in comparisons. The purpose of this analysis is to determine if our design choice of using unit content vectors (see Section 3.3) was a significant factor in our results.

To explore this question, we consider two variations on the content vector representation. The first variation is simply not to use any kind of normalization at all. That is, we simply use as the strength of each component of the content vector the number of statements and terms that contained the corresponding predicate. (The computation of normalized content vectors involves an additional step—dividing each component by the total number of predicates in the description.) The results of this manipulation are illustrated in Table 7. The key point to notice about this table is that the viable region is roughly the same size and shape as the viable region for normalized content vectors. This lends support to the claim that the outcome of the simulation experiments is not heavily determined by the particular normalization algorithm chosen.

The second variation we consider is to change what aspect of the overlap content vectors measure. Recall that the idea of content vectors is to compare the pattern of functors which appear in two structured descriptions. There are several ways to characterize such patterns. The MAC/FAC design choice, normalized content vectors, estimates the overlap in terms of the relative frequency of functors in the two descriptions, independent of their sizes. The unnormalized content vectors just examined estimate the total size of the overlap. But is it the pattern of overlap that is relevant, or just how many functors two descriptions have in common? We can investigate this question by changing the structure of content vectors so that they represent only the set of predicates that are used in the structured representation, without regard to number of occurrences. We call this variation *binary content vectors* because each component is essentially a 1 bit answer to the question of whether the structured representation contains or does not contain a particular predicate. Thus, the dot product of two binary content vectors is a measure of the overlap in number of shared predicates. (Again, we normalize to unit vectors, both to avoid size biases and because we assume that psychologically plausible implementation substrates (e.g., neural systems) will have limited dynamic range.) The results of this manipulation are shown in Table 8.



Rows: MAC widths. Columns: FAC widths.

	1%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100 %
1%	4	4	4	4	4	4	4	4	4	4	4
10%	4	Y	Y	Y	Y	Y	Y	Y	Y	256	256
20%	4	64	Y	Y	Y	Y	Y	256	256	256	256
30%	4	64	64	336	336	336	336	336	336	336	336
40%	4	64	Y	80	88	88	120	376	376	376	376
50%	4	64	64	80	80	80	112	368	368	368	368
60%	4	64	64	80	80	80	112	368	368	368	368
70%	4	64	64	80	80	80	112	368	368	368	368
80%	4	64	64	80	80	80	112	368	368	368	368
90%	4	64	64	80	80	80	112	368	368	368	368
100 %	4	64	64	80	80	80	112	368	368	368	368

Legend
4 = No analogies
64 = SF ≤ AN
80 = LS ≤ AN, SF ≤ AN
88 = LS ≤ SF, LS ≤ AN , SF ≤ AN
112 = LS ≤ AN, LS ≤ FA, SF ≤ AN
120 = LS ≤ SF, LS ≤ AN, LS ≤ FA, SF ≤ AN
256 = LS ≤ DT
336 = LS ≤ AN, SF ≤ AN, LS ≤ DT
368 = LS ≤ AN, LS ≤ FA, SF ≤ AN, LS ≤ DT
376 = LS ≤ SF, LS ≤ AN, LS ≤ FA, SF ≤ AN, LS ≤ DT

Rows are the width of the MAC selector, Columns correspond to the width of the FAC selector. The codes describe whether that combination of selector widths allows MAC/FAC to account for the human data, and if not, what criteria were violated.

Table 7. Sensitivity analysis, unnormalized content vectors.

Again, the overall pattern of results is the same: With selector widths that are too narrow, no analogies are retrieved, and with selector widths that are too broad, too many analogies are retrieved, followed as widths increase by too many "junk" retrievals. The interesting difference is that the region for the selector widths has changed: The viable wide-FAC range lies with MAC between 30% and 50%, whereas it was between 10% and 20% for the original content vectors. Comparing the average number of representations output by MAC for these ranges provides some insight as to why this should be so: For binary content vectors, the average output size was 2; for standard content vectors, the average was 1.5. In both cases, the next step of MAC selector width allows, on the average, another representation to make it through the FAC. Yet one more step in MAC selector width

Rows: MAC widths. Columns: FAC widths.

	1%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1%	4	4	4	4	4	4	4	4	4	4	4
10%	4	4	4	4	4	4	4	4	4	4	4
20%	4	4	4	4	4	4	4	4	4	4	4
30%	4	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
40%	4	Y	Y	64	Y	Y	Y	Y	Y	Y	Y
50%	4	64	Y	64	Y	Y	Y	Y	Y	Y	Y
60%	4	64	64	80	80	80	112	112	112	112	112
70%	4	64	64	80	80	80	112	112	368	368	368
80%	4	64	64	80	80	80	112	368	368	368	368
90%	4	64	64	80	80	80	112	368	368	368	368
100%	4	64	64	80	80	80	112	368	368	368	368

Legend
Y = Syslit predictions satisfied
4 = No analogies
64 = SF ≤ AN
80 = LS ≤ AN, SF ≤ AN
112 = LS ≤ AN, LS ≤ FA, SF ≤ AN
368 = LS ≤ AN, LS ≤ FA, SF ≤ AN, LS ≤ DT

Binary content vectors measure the size of overlap in predicates. As before, rows are the width of the MAC selector, Columns correspond to the width of the FAC selector. The codes describe whether that combination of selector widths allows MAC/FAC to account for the human data, and if not, what criteria were violated.

Table 8. Sensitivity analysis for binary content vectors.

allows many more representations to get through to FAC. Thus, measuring only the number of shared predicates shifts the viable region but does not substantially change its character.

From these two analyses, we conclude that the choice of normalization algorithm does not substantively affect the results. Because the normalization algorithm is not a theoretically determined choice, these analyses support the conclusion that the simulation works according to the theoretical account.

5.4 Sensitivity Analysis Three: Attributes Versus Relations

Content vectors homogenize structured representations. They unify information about attributes of objects, relationships between objects, and argument structure. Is including every kind of information in content vectors necessary? Given the frequency of literal-similarity and surface feature matches, both of which share many attributes, a possible hypothesis is that content vectors could be built using attributes alone. On the other extreme, the approaches used in case-based reasoning tend to ignore attributes and use only relational information. To mimic these approaches in MAC/FAC, we could use content vectors, which leave out attributes and include only relational predicates. This analysis explores both of these extreme hypotheses.

Rows: MAC widths. Columns: FAC widths.

	1%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1%	326	322	322	322	322	322	322	322	322	322	322
10%	326	336	336	336	336	336	336	336	336	336	336
20%	260	320	320	336	336	336	336	336	336	336	336
30%	4	64	320	336	344	344	344	344	344	344	344
40%	4	64	64	336	336	336	368	368	368	368	368
50%	4	64	64	80	88	88	120	376	376	376	376
60%	4	64	64	80	80	80	112	368	368	368	368
70%	4	64	64	80	80	80	112	368	368	368	368
80%	4	64	64	80	80	80	112	368	368	368	368
90%	4	64	64	80	80	80	112	368	368	368	368
100%	4	64	64	80	80	80	112	368	368	368	368

Legend
Y = Syslit predictions satisfied
4 = No analogies
64 = SF ≤ AN
80 = LS ≤ AN, SF ≤ AN
88 = LS ≤ SF, LS ≤ AN, SF ≤ AN
112 = LS ≤ AN, LS ≤ FA, SF ≤ AN
120 = LS ≤ SF, LS ≤ AN, LS ≤ FA, SF ≤ AN
60 = No analogies, LS ≤ DT 320 = SF ≤ AN, LS ≤ DT
322 = No surface matches, SF ≤ AN, LS ≤ DT
326 = No surface matches, no analogies, SF ≤ AN, LS ≤ DT
336 = LS ≤ AN, SF ≤ AN, LS ≤ DT
344 = LS ≤ SF, LS ≤ AN, SF ≤ AN, LS ≤ DT
368 = LS ≤ AN, LS ≤ FA, SF ≤ AN, LS ≤ DT
376 = LS ≤ SF, LS ≤ AN, LS ≤ FA, SF ≤ AN, LS ≤ DT

These results obtained are using content vectors which only included attributes, leaving out relations and logical connectives. As before, rows are the width of the MAC selector, Columns correspond to the width of the FAC selector. The codes describe whether that pair of selector widths allows MAC/FAC to account for the human data, and if not, what criteria were violated.

Table 9. Sensitivity analysis of attribute-only content vectors.

To explore the degree to which using attribute information only in content vectors would allow MAC/FAC to satisfy the data, we modified the algorithm which computes content vectors to ignore anything other than attributes. The results of the sensitivity analysis are shown in Table 9. The pattern of results is dramatically different than in previous experiments. There is no viable region at all. This experiment provides strong evidence that using attribute information alone in content vectors cannot satisfy the data.

The failure of attributes alone to provide adequate filtering may not be surprising. Is relational information alone enough? To explore this question we again modified the algorithm that computes content vectors, this time to not include attributes. These new content vectors, therefore, only contained relationships between objects and higher-order relations, such as logical connectives. The same methodology for the sensitivity analysis was followed.

Rows: MAC widths. Columns: FAC widths.

	1%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1%	260	260	260	268	268	268	268	268	268	268	268
10%	270	268	268	268	268	268	268	268	268	268	268
20%	262	320	256	256	384	384	384	384	384	384	384
30%	260	256	256	256	264	264	264	264	264	264	264
40%	4	64	Y	Y	Y	Y	Y	256	256	256	256
50%	4	64	Y	Y	Y	Y	Y	Y	256	256	256
60%	4	64	Y	64	80	80	80	336	336	336	336
70%	4	64	64	80	80	80	112	368	368	368	368
80%	4	64	64	80	80	80	112	368	368	368	368
90%	4	64	64	80	80	80	112	368	368	368	368
100%	4	64	64	80	80	80	112	368	368	368	368

Legend
Y = Satisfies the psychological data
4 = No analogies
64 = SF ≤ AN
80 = LS ≤ AN , SF ≤ AN
112 = LS ≤ AN , SF ≤ AN , LS ≤ FOR
256 = LS ≤ DT (260) = No analogies, LS ≤ DT, LS ≤ DT
262 = No surface matches, no analogies, LS ≤ DT, LS ≤ DT
264 = LS ≤ SF, LS ≤ DT, LS ≤ DT
268 = No analogies, LS ≤ SF, LS ≤ DT, LS ≤ DT
270 = No surface matches, no analogies, LS ≤ SF, LS ≤ DT, LS ≤ DT 3
36 = LS ≤ AN, SF ≤ AN, LS ≤ DT, LS ≤ DT
368 = LS ≤ AN , SF ≤ AN , LS ≤ FOR , LS ≤ DT
384 = AN ≤ FOR, LS ≤ DT

These results are obtained using content vectors which only included relations and logical connectives, leaving out attributes. As before, rows are the width of the MAC selector, Columns correspond to the width of the FAC selector. The codes describe whether that pair of selector widths allows MAC/FAC to account for the human data, and if not, what criteria were violated.

Table 10. Manipulation: Relation-only vectors.

The results of the sensitivity analysis are shown in Table 10. Like the attribute-only content vectors, the relation-only content vectors also fail to satisfy the data in a pyschologically plausible manner, but for different reasons. Almost uniformly, that is, when either the MAC width is less than 40% or when the FAC width is greater than 70%, more "junk" matches come through—stories from other sets, and FOR stories (i.e., those which match only in terms of first-order relations and not attributes or causal structure). The region where the data is not satisfied and the MAC width ranges between 20% and 70% is very much like the failures that occur for the attribute-only vectors (e.g., more analogies retrieved with narrow FAC than psychologically plausible). There is in fact a region in Table 10 where the pattern of results matches the human data, when the MAC width is between 40% and 50% and the FAC width ranges from 20% to either 60% or 70%. However, the size of the MAC output in this range is roughly one

Given a pool of memory items I1..In and a probe P:

1. For each item Ii, include it in a matching network if there are any predicates in Ii that are semantically similar to a predicate in P. The matching network implements semantic and structural constraints.
2. Create inhibitory links between units representing competing retrieval hypotheses, to ensure competitive retrieval.
3. Install pragmatic constraints by creating excitatory links between a special pragmatic node and every predicate marked by the user as important.
4. Run the network until it settles.

Figure 8. The ARCS algorithm

half of the total size of the memory pool. Consequently, this is not a viable region, because it demands far too much of FAC.

These experiments provide evidence that neither attribute information nor relational structure, by themselves, provide the right kind of information to allow the MAC/FAC model to plausibly satisfy the psychological data. Although such generalizations must be viewed with caution, the analysis of why these alternatives fail may be applied to any retrieval model, not just MAC/FAC. Using attribute information alone does not allow a retrieval system to satisfy the rare insights criterion, because the relational information is not used as a cue in retrieval. Using relational information alone tends to violate the scalability criterion, because large fractions of memory must be searched when the discrimination provided by the relational vocabulary is inadequate.

6. COMPARING MAC/FAC AND ARCS ON ARCS DATA SETS

As mentioned earlier, the model of similarity-based retrieval that is closest to MAC/FAC is ARCS (Thagard et al., 1990). The ARCS algorithm is shown in Figure 8. ARCS uses a localist connectionist network to apply semantic, structural, and pragmatic constraints to selecting items from memory. Most of the work in ARCS is carried out by the constraint satisfaction network, which provides an elegant mechanism for integrating the disparate constraints that Thagard et al. postulated as important to retrieval. The use of competition in retrieval is designed to reduce the number of candidates retrieved. Using pragmatic information provides a means for the system's goals to affect the retrieval process.

After the network settles, an ordering can be placed on nodes representing retrieval hypotheses based on their activation. Unfortunately, no formal criterion was ever specified by which a subset of these retrieval hypotheses is selected to be considered as what is retrieved by ARCS. Consequently, in the following experiments, we mainly focus on the subset of retrieval nodes mentioned by Thagard et al. (1990) in their article.

6.1 Theoretical Trade-Offs

Both models have their appeals and drawbacks. Here we briefly examine several of each.

- *Pragmatic effects:* In MAC/FAC, it is assumed that pragmatics and context affect retrieval according to what is encoded in the probe. That is, we assume that plans and goals are important enough to be explicitly represented and hence will affect retrieval. In ARCS, additional influence can be placed on particular subsets of such information by the user marking it as important. The trade-off between these alternatives will best be explored by embedding them in larger, task-oriented simulations, so we do not consider effects of pragmatics further in this article.
- *Utility of results:* Because MAC/FAC uses SME in the FAC stage, the result of retrieval can include novel candidate inferences. Because the purpose of retrieval is to find new knowledge to apply to the probe, this is a substantial advantage. ARCS could close this gap somewhat by using ACME (Holyoak & Thagard, 1989) as a postprocessor.
- *Initial filtering:* MAC/FAC's content vectors represent the overall pattern of predicates occurring in a structured description, so that the dot product cheaply estimates overlap. ARCS' commitment to creating a network if there is any predicate overlap places more of the retrieval burden on the expensive process of setting up networks. The inclusive rather than exclusive nature of ARCS' initial stage leads to the paradoxical fact that a system in which pragmatic constraints are central must ignore CAUSE, IF, and other inferentially important predicates to be tractable.
- *Modeling inter-item effects:* Wharton et al. (1994) have shown that ARCS can model effects of competition between memory items in heightening the relative effect of structural similarity to the probe.

Perhaps the most important issue is the notion of *semantic similarity.* A key issue in analogical processing is what criterion should be used to decide if two elements can be placed into correspondence. The FAC stage of MAC/FAC follows the standard structure-mapping position that *analogy is concerned with discovering identical relational systems.* Thus, other elements can be matched flexibly in service of relational matching: Any two entities can be placed in correspondence, and functions can be matched nonidentically if doing so enables a larger structure to match. But relations have only three choices: They can match identically, as in (a); they can fail to match, as in (b); if the surrounding structural match warrants it, they can be re-represented in such a way that part of their representation now matches identically, as in the shift from (c) to (d).

(a) HEAVIER [camel, cow]—HEAVIER [giraffe, donkey]
(b) HEAVIER [camel, cow]—BITE [dromedary, calf]

(c) HEAVIER [camel, cow]—TALLER [giraffe, donkey]
(d) GREATER [WEIGHT(camel), WEIGHT(cow)]—
GREATER [HEIGHT(camel), HEIGHT(cow)].

ACME and ARCS also share the intuition that analogy is a kind of compromise between similarity of larger structures and similarity of individual elements—*semantic similarity,* in Holyoak and Thagard's (1989) terms. But the total similarity metric is different. These systems use graded similarity at all levels and for all kinds of predicates; relations have no special status. Thus, ARCS and ACME might find pair (b) above more similar than pair (a), because of the object similarity. This would not be true for SME and MAC/FAC.

In ACME, semantic similarity was operationalized using similarity tables. For any potential matching term, a similarity table was used to assign a similarity rating, which was then combined with other evidence to decide whether the two predicates could match. Thus, in the examples above, both pair (b) and pair (c) stand a good chance of being matched, depending on the stored similarities between TALLER, HEAVIER, and BITE, camel, dromedary and giraffe, and so on.

In ARCS, an augmented subset of WordNet (Miller, Fellbaum, Kegl, & Miller, 1988) was used to make semantic similarity decisions. WordNet is a psycholinguistic database describing relationships between words. Two predicates in ARCS are considered semantically similar if their corresponding lexical concepts in WordNet are connected via links that denote particular relationships. The use of WordNet as a database for simple lexical inferences is an appealing idea. The lexical connections found in this way should have well-founded motivations. Nevertheless, it is important to remember that WordNet was intended as a lexicon, not a language of thought. Using the lexical concepts of WordNet as a predicate vocabulary requires assuming that there exist conceptual representations that correspond to these lexical concepts. That does not seem an implausible assumption. However, assuming that relationships between words, such as *synonym* or *antonym,* are used in the cognitive processing of internal representations seems implausible.

We prefer our tiered identicality account, which uses inexpensive inference techniques to suggest ways to re-represent nonidentical relations into a canonical representation language. Such canonicalization has many advantages for complex, rich knowledge systems, where meaning arises from the axioms in which predicates participate. When mismatches occur in a context where it is desirable to make the match, we assume that people make use of techniques of re-representation. An example of an inexpensive inference technique to suggest re-representation is Falkenhainer's (1987, 1990a) *minimal ascension* method, which looks for common superordinates when context suggests that two predicates should match. The use of pure identicality augmented by minimal ascension allowed Falkenhainer's PHINEAS system to model the discovery of a variety of physical theories by analogy. We believe that WordNet could be used in a similar fashion, because it has superordinate information.

Holyoak and Thagard (1989) have argued that broader (i.e., weaker) notions of semantic similarity are crucial in retrieval, for otherwise we would suffer from too many missed retrivals. Although this at first sounds reasonable, there is a counterargument based on memory size. Human memories are far larger than any cognitive simulation yet constructed. In such a case, the problem of false positives (i.e., too many irrelevant retrievals) becomes critical. False negatives are of course a problem, but they can be overcome to some extent by reformulating and re-representing the probe, treating memory access as an iterative process interleaved with other forms of reasoning (as in Lange & Warton's, 1992, 1993, REMIND model). Thus, it could be argued that strong semantic similarity constraints, combined with re-representation, are crucial in retrieval as well as in mapping.

How do these different accounts of semantic similarity fare in predicting patterns of retrival? In the rest of this section, we tackle this question by comparing the performance of MAC/FAC and ARCS on a variety of examples.

6.2 Computational Experiments Comparing MAC/FAC and ARCS

6.2.1 Methods

Each experiment below has a similar structure. First, each simulation is given a memory, consisting of one or more database drawn from the ARCS representations.[14] Then retrieval is tested with probes drawn from a small predefined set of stories, replicating Thagard et al.'s (1990) experiments. The memory a simulation operates over consists of one or more databases. In some cases, the memory is augmented by a particular story: for example, when probing with variant Hawk stories, the Thagard et al. encoding of the "Karla the Hawk" story is added to memory. (This is done to see if the retrieval system is able to find the base story amidst the distractors, given variations on the story as probes.)

For brevity, we specify the probe set and memory contents symbolically, using "/" to distingush probe set from memory and "+" to indicate set union. Thus, HAWK/(PLAYS + Karla Base) indicates an experiment where the database of plays was probed with the Hawk stories. A description of the data sets is used and these conventions is summarized in Figure 9.

Both MAC/FAC and ARCS take propositional representations as inputs, but their representation conventions are quite different. The most crucial

[14] To date we have been unsuccessful in getting ARCS to run on many of the representations we used in Sections 4 and 5. In some cases, ARCS' network does not settle after even 1,000 iterations, and run times of up to 12 h have been required.

Databases:
FABLES = 100 encodings of Aesop's fables, encoded by Thagard et.al.
PLAYS = 25 encodings of Shakespeare's plays, encoded by Thagard et.al.
Story sets used as probes and memory items:
HAWK = Thagard et.al.'s encoding of the "Karla the Hawk" story set, i.e., original story, analog, appearance match, false analogy, and literal similarity versions. Databases using these probes have the original story added to memory, except when the original story itself is used as a probe.
SG = Thagard et.al.'s encoding of the Sour Grapes fable plus variations, i.e., original story, analog, appearance, and literal similarity versions. Databases using these probes have the original story added to memory, except when the original story itself is used as a probe.
H&WSS = Thagard et.al's encoding of Hamlet and West Side Story. When Hamlet is used as a probe it is removed from memory. West Side Story is never placed in memory.
Convention: For convenience, we refer to an experimental setup by the probe stories followed by the database used, e.g., SG/(FABLES+PLAYS) means that the Sour Grapes fables were used as probes with a memory consisting of both plays and fables. When a story is used as a probe, it is removed from memory first.

Figure 9. Databases and experimental stories used in the experiments

difference is that structure-mapping treats attributes, relations, and functions differently, whereas ARCS does not distinguish them. We used the following rules in translation: (a) One-place predicates were classified as attributes, (b) multi-argument predicates were classified as relations, and (c) because the arguments to CAUSE could be either events or modal propositions, we treated predicates used as arguments to a CAUSE statement either as modal relations (e.g., BECOMING-TRUE) or functions (e.g., MARRIED, KILLED). Because functions can be substituted under structure-mapping's identicality criterion, we ran these experiments on representations translated both with and without rule (c), that is, with and without functions. With one exception, noted later, the results were essentially identical with either translation scheme.

All run times are measured according to the Lucid Common Lisp internal clock. A single computer[15] was used for both simulations, so that run times would be comparable.

Replication of computational experiments is still something of a novelty, and standards for ensuring that reported simulation results are repeatable have not yet been established in cognitive science. Nevertheless, we have taken many precautions to ensure that we have run ARCS correctly. Where numerical information was available, for instance, we matched numerical results reported by them to several decimal places. One concern was what should count as a retrival in ARCS. Neither the original ARCS paper nor the code defines a criterion for distinguishing when an item is actually retrieved (indeed, stories with negative activations were sometimes considered retrievals). In reporting ARCS results, we cut off the list of retrieved results where Thagard et al. (1990) did. In some cases (e.g., fables), this represented a sharp boundary, in other cases (e.g., plays), it did not.

[15] An IBM RS/6000 Model 530, with 128MB of RAM using Lucid Common Lisp 4.01.

ARCS results. Numbers in parantheses represent the level of activation computed by ARCS

Probe	Results	Seconds
Sour Grapes appearance	Sour Grapes (0.28)	120
Sour Grapes, analog	Sour Grapes (0.21)	81
Sour Grapes, literal similarity	Sour Grapes (0.25)	123

MAC/FAC Results. Numbers in parentheses represent the scores for that story.

Probe	Results	Seconds
Sour Grapes appearance	FAC: Sour Grapes (0.53) MAC: Sour Grapes (0.56)	0.3
Sour Grapes analog	FAC: Sour Grapes (2.03) MAC: Sour Grapes (0.62)	0.2
Sour Grapes literal similarity	FAC: Sour Grapes (2.03) MAC: Sour Grapes (0.62)	0.2

Table 11. Results for SG/Fables experiment.

6.3 Experiment 1: Sour Grapes Comparison

In the first study, the memory set consists of the fables, including the Sour Grapes fable, and the probes are variants of Sour Grapes. Table 11 shows the results. The results for ARCS match those reported for the simulation by Thagard et al. (1990). The MAC/FAC results are quite similar. Thus, both systems successfully retrieve Sour Grapes from a database of fables when given variations of it. However, MAC/FAC is substantially faster. The run-time difference is fairly typical; MAC/FAC tends to be two orders of magnitude faster than ARCS when tested with identical data on the same computer.

6.4 Experiment 2: Effects of Additional Memory Items on Retrieval (Soup Grapes)

To check the stability of results under changes in memory contents, we reran Experiment 1, adding the database of 25 Shakespeare plays encoded by Thagard et al. (1990) to the fables database. We then tested the simulations to see if they would retrieve Sour Grapes from the database of 125 fables and plays when probed with variations of Sour Grapes. The results are shown in Table 12. MAC/FAC's results remain unchanged, except for a small increase in processing time. ARCS, on the other hand, is distracted by

ARCS Results

Probe	Results	Seconds
Sour Grapes appearance	Sour Grapes (0.28)	327
Sour Grapes, analog	The Taming of the Shrew (0.22), Merry Wives (0.18), [11 stories], Sour Grapes (-0.19)	251
Sour Grapes, literal similarity	Sour Grapes (0.25)	373

MAC/FAC Results

Probe	Results	Seconds
Sour Grapes appearance	FAC: Sour Grapes (0.53) MAC: Sour Grapes (0.56)	0.4
Sour Grapes analog	FAC: Sour Grapes (2.03) MAC: Sour Grapes (0.62)	0.3
Sour Grapes, literal similarity	FAC: Sour Grapes (2.03) MAC: Sour Grapes (0.62)	0.3

Table 12. Results of SG probes, database = Fables + Plays.

the plays in one of the probe conditions. Increasing the memory by 25% has led to different results with ARCS. The results also hint at a possible size bias in ARCS: It appears to prefer larger descriptions in retrieval, at the cost of correct matches.

6.5 Experiment 3: Larger Probe Sizes

The results for MAC/FAC in Experiment 2 are satisfactory, however, ARCS' seemingly poor performance requires further investigation. Does the relative size of the probe matter in the memory swamping effect? To find this out, we again ran both simulations, first with the plays' database as memory, then with the 25 plays and 100 fables as memory, this time using as probes the Hamlet and West Side Story encodings, as represented by Thagard et al. (1990). Given Hamlet as a probe, the question is whether the systems can retrieve a tragedy, or at least another Shakespeare play. Given West Side Story as a probe, the challenge is more specific: to retrieve Romeo & Juliet, the analogous play.

Table 13 shows the results for plays only in memory, and Table 14 shows the results with both plays and fables in memory. The good news for ARCS is that the fables have only minimally intruded on the activation for the top-ranked retrieved plays. A Midsummer Night's Dream is ARCS' top-ranked retrieval for West Side Story, but it did also, as stated by Thagard et al. (1990), retrieve Romeo & Juliet.

MAC/FAC, on the other hand, only retrieves Romeo & Juliet with either probe. For West Side Story this is indeed the expected result (and we believe

ARCS results. Numbers in parentheses represent levels of activation for that item.

Probe	Results	Seconds
Hamlet	Romeo & Juliet (0.54), King Lear (0.53), Othello (0.46), Cymbeline (0.42), Macbeth (0.41), Julius Caesar (0.38)	1843
West Side Story	Midsummer Night's Dream (0.58), Romeo & Juliet (0.57)	2539

MAC/FAC results. Numbers in parentheses represent scores for that item.

Probe	Results	Seconds
Hamlet	FAC: Romeo & Juliet (6.79) MAC: Othello (0.86), Macbeth (0.85), Romeo & Juliet (0.83), Julius Caeser (0.81)	22
West Side Story	FAC: Romeo & Juliet (16.51) MAC: Romeo & Juliet (0.88)	13

Table 13. Results for Hamlet, West Side Story as probes, Plays database.

ARCS Results.

Probe	Results	Seconds
Hamlet	Romeo & Juliet (0.531), King Lear (0.528), Othello (0.45), Cymbeline (0.41), Macbeth (0.40), Julius Caesar (0.37)	4112
West Side Story	Midsummer Night's Dream (0.58), Romeo & Juliet (0.57)	5133

MAC/FAC Results

Probe	Results	Seconds
Hamlet	FAC: Romeo & Juliet (6.79) MAC: Othello (0.86), Macbeth (0.85), Romeo & Juliet (0.83), Julius Caesar (0.81), Fable52 (0.80)	26
West Side Story	FAC: Romeo & Juliet (16.51) MAC: Romeo & Juliet (0.88)	8

Table 14. Results for Hamlet, West Side Story as probes, Plays+Fables database.

more intuitive than ARCS' result), but what is happening with Hamlet? Examining the structural evaluation scores (e.g., the FAC scores) reveals that FAC considers the match between West Side Story and Romeo & Juliet to be excellent (16.51), which makes sense because the encodings of West Side Story and Romeo & Juliet have almost isomorphic structure. When Hamlet is the probe, FAC is relatively indifferent; the FAC scores were: Romeo & Juliet (6.79), Julius Caesar (5.49), Macbeth (3.72), Othello (2.67). The drop-off from Romeo & Juliet is 20%, which is below MAC/FAC's default cutoff of 10%.

ARCS Results

Probe	Results	Seconds
Karl, literal similarity	"Karla" base (0.67)	315
Karla appearance	Fable55 (0.4), [7 fables], "Karla" base (-0.17)	176
Karla, analogy	Fable23 (0.33), [7 fables], "Karla" base (-0.27)	127
Karla, first-order overlap	Fable23 (0.0907), Fable55 (0.0903), [13 fables], "Karla" base (-0.11)	17

MAC/FAC Results.

Probe	Results	Seconds
Karla, Literal Similarity	FAC: "Karla" (16.07) MAC: "Karla" (0.81), Fable71 (0.74)	6
Karla, apperance	FAC: "Karla" (7.92) MAC: "Karla" (0.71), Fable52 (0.71), Fable71(0.66), Fable27(0.65), Fable5(0.64)	7
Karla, analog	FAC: "Karla" (8.57) MAC: "Karla"(0.81), Fable52 (0.77), Fable5 (0.77), Fable71(0.76), Fable45(0.75), Fable59(0.75), Fable27(0.75)	14
Karla, First-order overlap	FAC: "Karla" (5.33), Fable5 (5.33) MAC: "Karla" (0.73), Fable71(0.71), Fable52(0.71), Fable5(0.71), Fable45(0.69), Fable59(0.68),Fable27(0.68)	7

Table 15. Results for HAWK probes, database=Fables+"Karla" base story.

6.6 Experiment 4: Hawk Stories

The goal of encoding the Hawk stories was to replicate the results of Karla the Hawk studies described in Section 2.1.1. Thagard et al. (1990) encoded one story set and used the relative activation levels of the stories computed by ARCS as relative retrieval probabilities for human subjects. As Section 4.4 pointed out, ARCS' order of retrieval was as follows: literal similarity, first-order overlap, appearance, analogy, which is not a close match to the observed human ordering of literal similarity, appearance, analogy, first-order overlap. By contrast, MAC/FAC matched the human ordinal results in our simulation of this experiment.

However, our purpose in this experiment was to pursue the question of stability of results under different distractors. We asked two questions: (a) Does MAC/FAC, using Thagard et al.'s (1990) encodings, perform appropriately, and (b) does changing the database used as ARCS' memory change its predicted outcomes? Both simulations were run with the Hawk stories as probes, with the fables (plus Karla story) as memory and with both fables and plays (plus the Karla story) as memory. The results are shown in Table 15 and Table 16, respectively.

ARCS Results.

Probe	Results	Seconds
"Karla", literal similarity	"Karla" base (0.67)	614
"Karla", appearance	Fable55 (0.40),[16 stories], "Karla" base (-0.018)	408
"Karla", analogy	Pericles (0.60), [17 stories], "Karla" base (-0.32)	244
"Karla", false analogy	Pericles (0.58), [22 stories], "Karla" base (-0.38)	45

MAC/FAC Results.

Probe	Results	Seconds
Karla, Literal Similarity	FAC: "Karla"(16.07) MAC: "Karla"(0.81), Fable71 (0.74)	7
Karla, apperance	FAC: "Karla" (7.92), MAC: "Karla" (0.71), Fable52(0.71), Julius Caesar (0.69), Othello (0.68), Macbeth (0.67), Fable71(0.66), Two Gentlemen of Verona (0.65), Fable27(0.65), Hamlet (0.65), Fable5(0.64)	21
Karla, analog	FAC:"Karla"(8.57) MAC: "Karla" (0.81), Julius Caesar (0.78), Two Gentlemen of Verona (0.78), Fable52(0.77), Fable5(0.77), Macbeth (0.76),As You Like It(0.76), Fable71(0.76), Fable45(0.75), Fable59(0.75), Fable27(0.75), Othello(0.75)	37
Karla, First-order overlap	FAC: "Karla"(5.33), Fable5(5.33) MAC: "Karla"(0.73), Juilius Caesar(0.72), Two Gentlemen of Verona (0.72), Fable71(0.71), Fable52(0.71), Fable5 (0.71), Macbeth(0.70), As You Like It (0.70), Othello (0.69), Fable45 (0.69), Hamlet(0.68)	23

Table 16. Results for HAWK probes, with database=Fables+Plays+"Karla" base story.

No matter which database is used, MAC/FAC always retrieves the Karla story, irrespective of which variant story is used as a probe. The MAC scores explain why: In each case, the Karla story is at the top of the ranking, indicating that the pattern of identical predicates overlapping is greater for Karla and variant than for any other story. The fact that the Karla base story is retrieved for the literal similarity and appearance variants is expected. Its retrieval when the analogy is used as a probe is also reasonable (although if ARCS always retrieved analogs successfully it would be an implausible model). Retrieving the base story when the first-order overlap story is used as a probe is not so reasonable. We believe this occurs because the Thagard et al. (1990) representations are rather sparse and include almost no surface information and, thus, are less natural than might be desired (cf. the *specificity conjecture* of Forbus & Gentner, 1989).

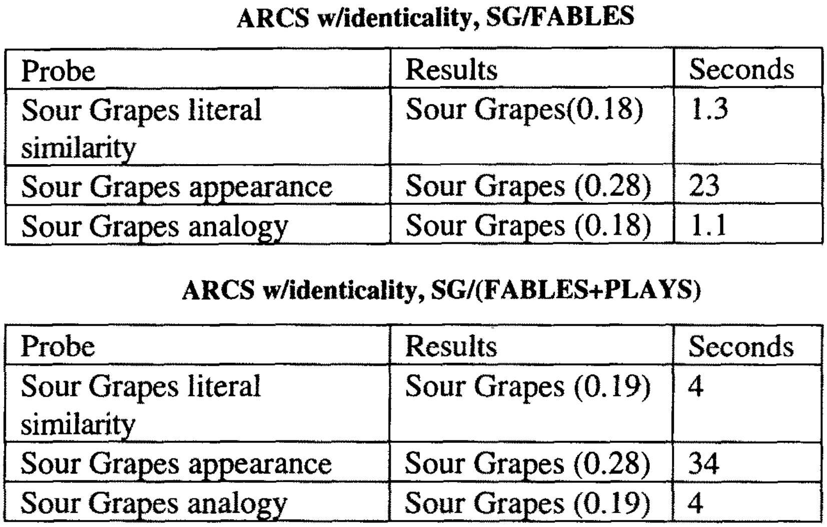
Interestingly, this experiment marks the only place where the decision to use functions in encoding made any real difference in the results. If no functions were used in translating the ARCS representations, the MAC results remained the same (because content vectors are based strictly on identical predicates), but the Karla base story would be knocked out of the FAC output by other stories that had more overlapping structure, because the causal structure in the Karla story could not be consistently mapped due to nonidentical relations. The fact that this problem only shows up with this one probe set, out of all the representations made by Thagard et al. (1990), suggests that this is not a serious problem.

As was suggested by Experiments 1 and 2, the ARCS results vary considerably with different distractor sets. This means that the use of relative activations to estimate relative frequencies is not a stable measure. Specifically, the relative ordering of first-order overlap and analogy reverses when the database of fables is augmented with the plays. The position of the Karla story in the activation rankings is also alarming. The appearance story, which should retrieve the base almost as often as the literal similarity story, has dropped from the 9th in the ranking to 18th. Depending on where the retrieval cutoff is placed, the conclusion might be that ARCS fails to retrieve the Karla story given the very close surface match.

6.7 Experiment 5: ARCS Using Simple Identicality

The results so far indicate that MAC/FAC is far more immune to false positives than ARCS. What is responsible for this difference? Is it MAC/FAC's use of a separate stage that performs structural filtering? The use of content vectors versus parallel constraint satisfaction to generate an initial set of retrieval candidates? MAC/FAC's identicality constraint versus ARCS' weaker semantic constraint? A complete answer to this question will require much more empirical and theoretical work, but we can gain some insight by a simple experiment. We ran ARCS again, but without the WordNet-inspired similarity network. Under such conditions, ARCS only creates local matches between identical predicates, and the initial candidate set is much smaller, because the semantic similarity constraint has been greatly tightened.

The results of this experiment are shown in Tables 17 through 19. Table 17 shows that the results on Sour Grapes have improved substantially; ARCS is no longer tempted by plays. Table 18 shows that, although a Midsummer Night's Dream is high on ARCS' list, it no longer prefers it to Romeo & Juliet when West Side Story is used as a probe. The Hawk results show the least improvement; the estimated retrieval order again does not match that of human participants, and there are still many fables and plays ahead of what should be very close matches to the Karla base story.



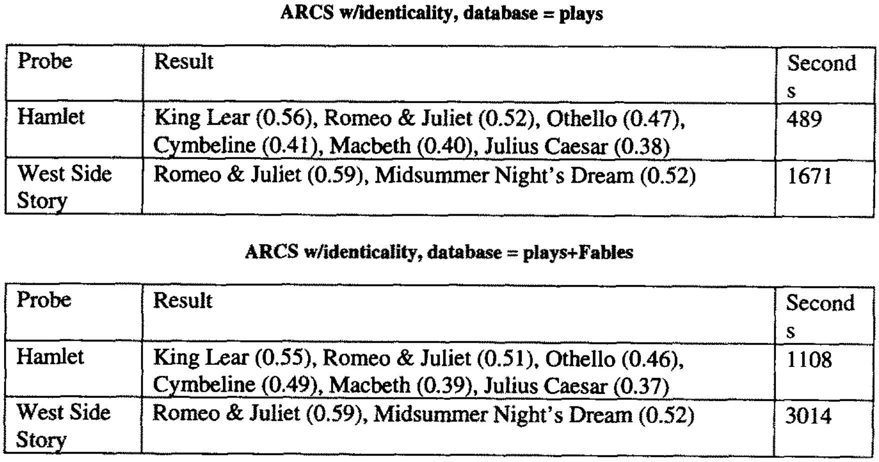
ARCS w/identicality, SG/FABLES

Probe	Results	Seconds
Sour Grapes literal similarity	Sour Grapes(0.18)	1.3
Sour Grapes appearance	Sour Grapes (0.28)	23
Sour Grapes analogy	Sour Grapes (0.18)	1.1

ARCS w/identicality, SG/(FABLES+PLAYS)

Probe	Results	Seconds
Sour Grapes literal similarity	Sour Grapes (0.19)	4
Sour Grapes appearance	Sour Grapes (0.28)	34
Sour Grapes analogy	Sour Grapes (0.19)	4

Table 17. ARCS w/identicality on Sour Grapes with Fables and Fables+Plays.



ARCS w/identicality, database = plays

Probe	Result	Seconds
Hamlet	King Lear (0.56), Romeo & Juliet (0.52), Othello (0.47), Cymbeline (0.41), Macbeth (0.40), Julius Caesar (0.38)	489
West Side Story	Romeo & Juliet (0.59), Midsummer Night's Dream (0.52)	1671

ARCS w/identicality, database = plays+Fables

Probe	Result	Seconds
Hamlet	King Lear (0.55), Romeo & Juliet (0.51), Othello (0.46), Cymbeline (0.49), Macbeth (0.39), Julius Caesar (0.37)	1108
West Side Story	Romeo & Juliet (0.59), Midsummer Night's Dream (0.52)	3014

Table 18. ARCS w/identicality, probed with Plays.

6.8 Conclusions from Computational Comparison Experiments

The results of cognitive simulation experiments must always be interpreted with care. In this case, we believe our experiments provide evidence that MAC/FAC, using structure-mapping's identicality constraint, better models retrieval than ARCS, which uses Thagard et al.'s (1990) notion of semantic

ARCS w/identicality, HAWK/FABLES

Probe	Results	Seconds
Karla, Literal Similarity	Fable23(0.261),Fable55(0.258),Karla story (-0.1)	73
Karla, Appearance	Fable55(0.4),[8 fables],Karla story (-0.23)	114
Karla, True Analogy	Fable23(0.26),Fable55(0.26),[5 fables], Karla story (-0.23)	12
Karla, First-Order overlap	Fable23(0.087),Fable55(0.087)	5

ARCS w/identicality, HAWK/(FABLES+PLAYS)

Probe	Results	Seconds
Karla, Literal Similarity	Fable23(0.26),Fable55(0.26),Karla story(-0.014)	74
Karla, Appearance	Fable55(0.25),Hamlet(0.17),Fable23(0.067),[17 plays & fables],Karla story(-0.22)	154
Karla, True Analogy	Pericles(0.55),[3 plays],Fable23(-0.13),Fable55(-0.13), [8 plays & fables],Karla story (-0.30)	29
Karla, First-Order overlap	Pericles(0.59),[6 fables & plays],Fable23(-0.25), Fable55(-0.25)	18

Table 19. ARCS w/identicality, HAWK probes.

similarity. In retrieval, the special demands of large memories argue for simpler algorithms, simply because the cost of false positives is much higher. If retrieval were a one-shot, all-or-nothing operation, the cost of false negatives would be higher. But that is not the case. In normal situations, retrieval is an iterative process, interleaved with the construction of the representations being used. Thus, the cost of false negatives is reduced by the chance that reformulation of the probe, due to re-representation and inference, will substantially catch a relevant memory that slipped by once.

Overall, although both MAC/FAC and ACME are designed to allow parallel implementations, MAC/FAC's speed advantage (roughly two orders of magnitude) would suggest that it is the more practical choice for cognitive simulation experiments. Finally, we note that although ARCS' use of a localist connectionist network to implement constrain satisfaction is in many ways intuitively appealing, it is by no means clear that such implementations are neurally plausible. On the other hand, we believe the evidence suggests that MAC/FAC captures similarity-based retrieval phenomena better than ARCS does.

7. DISCUSSION

To understand the role of similarity in transfer requires making fine distinctions both about similarity and about transfer. The psychological evidence indicates that the accessibility of matches from memory is strongly influenced by surface commonalities and weakly influenced by structural commonalities, whereas the rated inferential soundness of comparisons is strongly influenced by structural commonalities and is little, if at all, influenced by surface commonalities. An account of similarity in transfer must deal with the dissociation between retrieval and structural alignment: between the matches people *get* from memory and the matches they *want*.

The MAC/FAC model of similarity-based retrieval captures both the fact that humans successfully store and retrieve intricate relational structures and the fact that access to these stored structures is heavily (though not entirely) surface driven. The first stage is attentive to content and blind to structure, and the second stage is attentive to both content and structure. The MAC stage uses content vectors, a novel summary of structured representations, to provide an inexpensive "wide net" search of memory, whose results are pruned by the more expensive literal similarity matcher of the FAC stage to arrive at useful, structurally sound matches.

The simulation results presented here demonstrate that MAC/FAC can simulate the patterns of access exhibited by humans. It displays the appropriate preponderance of literal similarity and surface matches, and it occasionally retrieves purely relational matches (Section 4). Our sensitivity studies suggest that these results are a consequence of our theory and are not hostage to nontheoretically motivated parameters or algorithmic choices (Section 5). Our computational experiments comparing MAC/FAC and ARCS (Section 6) suggests that MAC/FAC accounts for the psychological results more accurately and more robustly than ARCS. In addition to the experiments reported here, we have tested MAC/FAC on a variety of other data sets, including relational metaphors (30 descriptions, average of 12 propositions each) and attribute-rich descriptions of physical situations as might be found in commonsense reasoning (12 descriptions, averaging 42 propositions each). We have also tried various combinations of these databases with the Karla the Hawk data set (45 descriptions, averaging 67 propositions each). In all cases to date, MAC/FAC's performance has been satisfactory and consistent with the overall pattern of findings regarding human retrieval. We conclude that MAC/FAC's two-stage retrieval process is a promising model of human retrieval.

7.1 Limitations and Open Questions

7.1.1 Retrieval Failure

Sometimes a probe reminds us of nothing. Currently the only way this can happen in the MAC/FAC model is for FAC to reject every candidate provided by MAC. This can happen if no structurally sound match hypotheses can be generated between the probe and the descriptions output by MAC.

(Without any local correspondences there can be no interpretation of the comparison.) This can happen, albeit rarely. A variant of MAC/FAC with thresholds on the output of either or both MAC or FAC stages—so that the system would return nothing if the best match were below criterion—would show more nonremindings.

7.1.2 Focused Remindings and Penetrability

Many AI retrieval programs and cognitive simulations elevate the reasoner's current goals to a central role in their theoretical accounts (e.g., Burstein, 1989; Hammond, 1986; 1989; Keane, 1988a, 1989b; Kolodner, 1984, 1989; Riesbeck & Shank, 1989; Thagard et al., 1990). Although we agree with the claim that goal structures are important, MAC/FAC does not give goals a separate status in retrieval. Rather, we assume that the person's current goals are represented as part of the higher-order structure of the probe. The assumption is that goals are embedded in a relational structure linking them to the rest of the situation; they play a role in retrieval, but the rest of the situational factors must participate as well. When one is hungry, for instance, presumably the ways of getting food that come to mind are different if one is standing in a restaurant, a supermarket, or in the middle of a forest. The inclusion of current goals as part of the representation of the probe is consistent with the finding of Read and Cesa (1991) that asking subjects for explanations of current scenarios leads to a relatively high rate of analogical reminding. However, we see no reason to elevate goals above other kinds of higher-order structure. By treating goals as just one of many kinds of higher-order structures, we escape making the erroneous prediction of many case-based reasoning systems: that retrieval requires common goals. People can retrieve information that was originally stored under different goal structures. (See Goldstein, Kedar, & Bareiss, 1993, for a discussion of this point.)

A related question concerns the degree to which the results of each stage are inspectable and tunable. We assume that the results of the FAC stage are inspectable, but that explicit awareness of the results of the MAC stage is lacking. We conjecture that one can get a sense that there are possible matches in the MAC output, and perhaps some impression of how strong the matches are, but not what those items are. The feeling of being reminded without being able to remember the actual item might correspond to having candidates generated by MAC that are all either too weak to pass on or are rejected by the FAC stage. Some support for this two-stage account comes from Metcalfe's (1993) findings on feeling-of-knowing. She found that subjects report a general sense of feeling-of-knowing *before* they can report a particular retrieval. Reder (1988) suggests that this preretrieval sense of feeling-of-knowing might provide the basis for deciding whether to pursue and expect retrieval.

This raises the question of how much MAC and FAC can be affected by the subject? There is psychological evidence that people cannot directly control the kinds of matches they retrieve. Schumacher and Gentner (in press) investigated this by varying the test instructions given to subjects. They gave subjects lists of proverbs to read, followed by test proverbs which were either structurally similar or surface-similar to proverbs studied previously. Subjects who were told to write any prior proverbs that they were reminded of while reading the test proverbs recalled about twice as many surface matches as analogies. Another group of subjects was told to write only structural remindings and to strive for as many of these as possible. Although these subjects indeed wrote many fewer surface matches than the first group, they recalled only the same low number of analogies. The goal to seek relational matches apparently led people to filter nonrelational matches, but not to find more relational matches. This suggests that the FAC matcher may be tunable (in that subjects were able to filter out the surface matches) but not the MAC matcher (in that subjects were not able to produce more analogies on demand).

The idea that FAC, though not MAC, is tunable is consistent with evidence that people can be selective in similarity matching once both members of a pair are present. For example, in a triads task, matching XX to OO or XO, subjects can readily choose either only relational matches (XX—OO) or only surface matches (XX—XO) (Gentner & Markman, 19994a, 1994b; Goldstone et al., 1991; Medin et al., 1993). This kind of structural selectivity in the similarity processor is readily modeled in SME (by assuming that we select the interpretation that fits the task constraints), but not in ACME (Holyoak & Thagard, 1989). ACME produces one best output that is its best compromise among the current constraints. It can be induced to produce different preferred mappings by inputting different pragmatic activations, but not by inviting different structural preferences (Spellman & Holyoak, 1992).

7.1.3 Size of Content Vectors

One potential problem with scaling up with MAC/FAC is the potential growth in the size of content vectors. Our current descriptions use a vocabulary of only a few hundred distinct predicates. We implement content vectors via sparse encoding techniques, analogous to those used in computational matrix algebra, for efficiency. However, a psychologically plausible representation vocabulary may have hundreds of thousands of predicates. It is not obvious that our sparse encoding techniques will suffice for vocabularies that large, nor does this implementation address the question of how systems with limited "hardware bandwidth," such as connectionist implementations, could serve as a substrate for this model.

This scale problem is mitigated partly by MAC/FAC's basic architecture with its cheap initial filter. However, there are at least two further possible ways to address the potential scale problem in the size of content vectors. The first is *abstraction.* In symbolic knowledge representations, predicates and functions are often arranged in hierarchies. For example, a complex concept such as *bequeath* might be stored as a specialization of the concept of *giving,* which might in turn be a specialization of the concept of *transfer.* Let us view the set of specializations between predicates as a lattice. Any set of predicates that partitions the lattice can be used to formulate a *semantically compressed* content vector as follows: The weight of a component of the compressed content vector is a function of the number of occurrences of that predicate—and all predicates below it in the partition—in the description. In effect, predicates below the selected subsets are replaced with more abstract versions. Another possible solution for the scale problem is *factorization:* The predicates could be partitioned into subsets that are tightly interrelated, and separate content vectors could be computed for each subset. This organization presumes that there is some fixed size bound on processing modules, but that several processing modules can be synchronized well enough to accumulate results across them.

7.1.4 Combining Similarity Effects Across Items

MAC/FAC is currently a purely exemplar-based memory system. The memory items can be highly situation-specific encodings of perceptual stimuli, abstract mathematical descriptions, causal scenarios, and so forth. MAC/FAC lacks the capacity to model inter-item effects. For example, MAC/FAC does not capture competition among items. Wharton, Holyoak, Downing, Lange, and Wickens (1991, 1992) and Wharton et al. (1994) have shown an intriguing effect where competition between exemplars heightens the relative effect of structural similarity in retrieval. MAC/FAC also does not average across several items at retrieval (Medin & Schaffer, 1978) or derive a global sense of familarity by combining the activations of multiple retrievals (Gillund & Shiffrin, 1984; Hintzman, 1986, 1988). An interesting extension of MAC/FAC would be to include this kind of between-item processing upon retrieval.

If such inter-item averaging occurs, it could provide a route to the incremental construction of abstractions and indexing information in memory. We see three plausible ways to do this. First, as above, the descriptions output by the MAC stage could be compared. Second, the access system might incrementally build up something like Minsky's (1981) similarity network, using the history of retrievals to encode difference descriptions to simplify future access. Third, the descriptions output by the FAC stage could be compared: SME could be used to carry out structural abstraction across several descriptions (as in Skorstad et al., 1988) to produce a combined description as the FAC output. The first and third models are both forms of "late averaging" accounts, and it would be interesting to compare these techniques with other models that account for prototype effects by combining exemplars at retrieval (Hintzman, 1986, 1988; Medin & Shaffer, 1978).

7.1.5 Iterative Access

Keane (1988c, 1991; Keane & Brayshaw, 1988) and Burstein (1983a, 1983b) have proposed incremental mapping processes. We suggest that similarity-based retrieval may also be an iterative process. In particular, in active retrieval (as opposed to spontaneous remindings), we conjecture that MAC/FAC may be used iteratively, each time modifying the probe in response to the previous match (cf. Falkenhainer, 1987, 1990a; Gentner, 1989). Suppose, for example, a probe yielded several partial remindings. The system of matches could provide clues as to which aspects of the probe are more or less relevant and, thus, should be highlighted or suppressed on the next iteration. MAC should respond to this altered vector by returning more relevant items, and FAC can then select the best of these.

Another advantage of such incremental reminding is that it might help explain how we derive new relational categories. Barsalou's (1982, 1987) *ad hoc* categories, such as "things to take on a picnic" and Glucksberg and Keysar's (1990) metaphorically based categories, such as "jail" as a prototypical confining institution, are examples of the kinds of abstract relational commonalities that might be highlighted during a process of incremental retrieval and mapping.

7.1.6 Embedding in Performance-Oriented Models

MAC/FAC is not itself a complete analogical processing system. For example, both constructing a model from multiple analogs (e.g., Burstein, 1983a, 1983b) and learning a domain theory by analogy (e.g., Falkenhainer, 1987, 1988, 1990b) require multiple iterations of accessing, mapping, and evaluating descriptions. Several psychological questions about access cannot be studied without embedding MAC/FAC in a more comprehensive model of analogical processing (Forbus & Gentner, 1991). First, as discussed previously, there is ample evidence that subjects can choose to focus on different kinds of similarity when the items being compared are both already in working memory. Embedding MAC/FAC in a larger system should help make clear whether this penetrability should be modeled as applying to the FAC system or to a separate similarity engine. (Order effects in analogical problem solving [Keane, 1990] suggest the latter.)

A second issue that requires a larger, performance-oriented model to explore via simulation is when and how pragmatic constraints should be incorporated (cf. Holyoak & Thagard, 1989; Thagard & Holyoak, 1989, 1990). Because we assume that goals, plans, and similar control knowledge is explicitly represented in working memory, the MAC stage will include such predicates in the content vector for the probe and hence will be influenced by pragmatic concerns. There are two ways to model the effects of pragmatics on the FAC stage. The first is to use the SME *pragmatic marking algorithm* (Forbus & Oblinger, 1990) as a relevance filter. The second is to use incremental mapping, as in Keane and Brayshaw's (1988) Incremental Analogy Machine (IAM). This technique permits the selection and grouping of sets of correspondences to be influenced by the task at hand (Forbus et al., 1994).

A recent simulation by Lange and Wharton (1992, 1993), REMIND, models retrieval in the context of natural language processing, using spreading activation in a connectionist network both to construct a conceptual representation from textual input and to find the most similar story in its episodic memory. REMIND is an intriguing model, and the attempt to integrate multiple cognitive processes into larger models is an important activity. However, it is difficult to compare this model with MAC/FAC and other retrieval models. First, REMIND only models a specific retrieval task, namely retrieval in the service of understanding stories, and thus does not attempt to cover as wide a span of phenomena as MAC/FAC. Second, when REMIND retrieves a story, it does not appear to create correspondences between the understanding of its input and the previous story, nor does it generate novel candidate inferences, and thus does not satisfy the structured mappings criterion for retrieval. Third, REMIND has only been tested on a corpus involving a handful of short (i.e., two sentence) stories. To our knowledge, it has never been tested either on a corpus as large as those used with MAC/FAC and ARCS or on a corpus that includes examples as large as those used with MAC/FAC. Even their current small databases stretch the limits of a Connection Machine,[16] which makes it difficult to evaluate their model thoroughly.

7.1.7 Expertise and Relational Access

Despite the gloomy picture painted in this research and in most of the problem-solving research, there is evidence of considerable relational access (a) for experts in a domain and (b) when intial encoding of the study set is relatively intensive. Novick (1988a, 1988b) studied remindings for mathematics problems using novice and expert mathematics students. She found that experts were more likely than novices to retrieve a structurally similar prior problem, and when they did retrieve a surface-similar problem, they were quicker to reject it than were novices. Faries and Reiser (1988) taught participants LISP in a series of intensive training sessions and then gave them target problems that were superfically similar to one prior problem and structurally similar to another. Given this intensive training, Faries and

[16] Trent Lange, personal communication, IJCAI-93.

Reiser's subjects were able to access structurally similar problems despite the competing superficial similarities.

The second contributor to relational retrieval, almost certainly related to the first, is intensive encoding. Gick and Holyoak (1983) and Catrambone and Holyoak (1987, 1989) found that subjects exhibited increased relational retrieval when they were required to compare two prior analogs, but not when they were simply given two prior analogs to read. Schumacher and Gentner (1987) found increased relational retrieval of proverbs when subjects wrote out the meaning of each proverb on the study list, as opposed to simply reading it or rating its cleverness. Seifert, McKoon, Abelson, and Ratcliff (1986) investigated priming effects in a sentence verification task between thematically similar (analogical) stories. They obtained priming when subjects first studied a list of themes and then judged the thematic similarity of pairs of stories, but not when subjects simply read the stories.

The increase of relational reminding with expertise and with intensive encoding can be accommodated in the MAC/FAC model. First, we assume that experts have richer and better structured representations of the relations in the content domain than do novices (Carey, 1985; Chi, 1978; Reed, Ackinclose, & Voss, 1990). This fits with developmental evidence that as children come to notice and encode higher-order relations such as *symmetry* and *monotonicity,* their appreciation of abstract similarity increases (Gentner & Rattermann, 1991; Kotovsky & Gentner, 1990). Second, in particular we speculate that experts may have a more uniform internal relational vocabulary within the domain of expertise than do novices (Clement, Mawby, & Giles, 1994; Gentner & Rattermann, 1991; Gentner, Rattermann, Kotovsky, et al., in press). The idea is that experts tend to have relatively comprehensive theories in a domain and that this promotes canonical relational encodings within the domain.

To the extent that a given higher-order relational pattern is used to encode a given situation, it will of course be automatically incorporated into MAC/FAC's content vector. This means that any higher-order relational concept that is widely used in a domain will tend to increase the uniformity of the representations in memory. This increased uniformity should increase the mutual accessibility of situations within the domains. Thus, as experts come to encode a domain according to a uniform set of principles, the likelihood of appropriate relational remindings increases. That is, under the MAC/FAC model, the differences in retrieval patterns for novices and experts are explained in terms of differences in knowledge, rather than by the construction of explicit indices.

Bossok has made an interesting argument that indirectly supports this claim of greater relational uniformity for experts than for novices (Bassok & Wu, in press). Noting prior findings that in forming representations of novel texts people's interpretations of verbs depend on the nouns attached to them (Gentner, 1981; Gentner & France, 1988), Bassok suggests that particular representations of the relational structure may thus be idiosyncratically related to the surface content, and that this is one contributor to the poor relational access. If this is true, and if we are correct in our supposition that experts tend to have a relatively uniform relational vocabulary, then an advantage for experts in relational access would be predicted.

As domain expertise increases, MAC/FAC's activity may come to resemble a multigoal case-based reasoning model with complex indices (e.g., Birnbaum & Collins, 1989; King & Bareiss, 1989; Martin, 1989; Pazzani, 1989; Porter, 1989). We can think of its content vectors as indices with the property that they change automatically with any change in the representation of domain exemplars. Thus, as domain knowledge—particularly the higher-order relational vocabulary—increases, MAC/FAC may come to have sufficiently elaborated representations to permit a fairly high proportion of relational remindings. The case-based reasoning emphasis on retrieving prior examples and generalizations that are inferentially useful may thus be a reasonable approximation to the way experts retrieve knowledge.

Although MAC/FAC's two-stage operation is not generally shared by case-based models, it is shared by one case-based reasoning system that uses a two-stage model, the CaPER system (Kettler, Hendler, & Anderson, 1992). CaPER is designed to retrieve all sufficiently similar plans from an unindexed case base, beginning with a massively parallel stage which does a simple, nonstructural match between a query and the contents of memory. It would be very interesting to see how well the parallel techniques used in CaPER could be applied to MAC/FAC.

7.2 The Decomposition of Similarity

The dissociation between surface similarity and structural similarity across different processes has broader implications for cognition and is related to several recent discussions. Medin et al. (1993) and Gentner (1989) have argued that similarity is pluralistic, in the sense that there are multiple subclasses of similarity and multiple influences on how it is computed. Rips (1989) demonstrated a dissociation between similarity, typicality, and categorization. Murphy and Medin (1985) and Keil (1989) have commented on the limited usefulness of simple similarity and pointed out that physical resemblance does not provide a sufficient basis for determining conceptual groupings. As discussed above, there is a relational shift in development (Gentner & Rattermann, 1991; Halford, 1992). Finally, local object matches appear to be processed faster by adults than structural commonalities. Goldstone and Medin (1994a, 1994b) found that local similarities have their effects on mapping earlier than global relational similarities in a timed mapping task, and Ratcliff and McKoon (1989) found convergent results in a sentence-matching task: Subjects could discriminate new from old sentences faster if the the new sentences contained all new words (e.g., "Helen attracted Jeff." vs. "Andrew accosted Mary.") than if the sentences differed only in relational structure (e.g., "Helen attracted Jeff." vs. "Jeff attracted Helen."). In pilot experiments using perceptual stimuli, in which subjects were timed under different kinds of mapping instructions, Markman and Gentner (in press) found that subjects are faster to choose on the basis of similar objects than on the basis of similar relations, even when the two rules dictate the same response.

These kinds of results render less plausible the notion of a unitary similarity that governs retrieval, evaluation, and inference. Instead, they suggest a more complex, pluralistic view of similarity. MAC/FAC provides an architecture that demonstrates how such a pluralistic notion of similarity can be organized to account for psychological data on retrieval.

REFERENCES

Barnden, J.A. (1994). On the connectionist implementation of analogy and working memory matching. In K.J. Holyoak & J.A. Barnden (Eds.), *Advances in connectionist and neural computation theory, Vol. 2: Analogical connections* (pp. 327-374). Norwood, NJ: Ablex.

Barsalou, L.W. (1982). Context-independent and context-dependent information in concepts. *Memory and Cognition, 10,* 82-93.

Barsalou, L.W. (1987). The instability of graded structure: Implications for the nature of concepts. In U. Neisser (Ed.), *Concepts and conceptual development: Ecological and intellectual factors in categorization.* Cambridge, England: Cambridge University Press.

Bassok, M., Wu, L., & Olseth, K. L. (1995). Judging a book: Interpretative effects of content on problem solving transfer. *Memory & Cognition, 23*(3), 354-367.

Birnbaum, L., & Collins, G. (1989). Remindings and engineering design themes: A case study in indexing vocabulary. *Proceedings: Case-Based Reasoning Workshop* (pp. 47-51). San Mateo, CA: Morgan Kaufmann.

Branting, K.L. (1991). Building explanations from rules and structured cases. *International Journal of Man-Machine Systems.*

Burstein, M.H. (1983a). Concept formation by incremental analogical reasoning and debugging. *Proceedings of the International Machine Learning Workshop* (pp. 19-25). Monticello, Urbana: University of Illinois.

Burstein, M.H. (1983b). A model of learning by incremental analogical reasoning and debugging. *Proceedings of the National Conference on Artificial Intelligence* (pp. 45-48). Washington, DC, Los Altos, CA: Morgan Kaufmann.

Burstein, M. (1989). Analogy vs. CBR: The purpose of mapping. *Proceedings: Case-Based Reasoning Workshop* (pp. 133-136). San Mateo, CA: Morgan Kaufmann.

Carey, S. (1985). *Conceptual change in childhood.* Cambridge, MA: MIT Press.

Catrambone, R., & Holyoak, K.J. (1987, May). *Do novices have schemas?* Paper presented at the fifty-second annual meeting of the Midwestern Psychological Association, Chicago, IL.

Catrambone, R., & Holyoak, K.J. (1989). Overcoming contextual limitations on problem-solving transfer. *Journal of Experimental Psychology: Learning, Memory and Cognition, 15.*

Chi, M.T.H. (1978). Knowledge structures and memory development. In R.S. Siegler (Ed.), *Children's thinking: What develops?* (pp. 73-96). Hillsdale, NJ: Erlbaum.

Clement, J. (1986). Methods for evaluating the validity of hypothesized analogies. *Proceedings of the eighth annual Conference of the Cognitive Science Society* (pp. 223-234). Amherst, MA, Hillsdale, NJ: Erlbaum.

Clement, C.A., & Gentner, D. (1991). Systematicity as a selection constraint in analogical mapping. *Cognitive Science, 15,* 89-132.

Clement, C.A., Mawby, R., & Giles, D.E. (1994). The effects of manifest relational similarity on analogy retrieval. *Journal of Memory and Language, 33,* 396-420.

Falkenhainer, B. (1987, August). *An examination of the third stage in the analogy process: Verification-based analogical learning. Proceedings of the tenth International Joint Conference on Artificial Intelligence* (pp. 260-263). Milan, Italy, Los Altos, CA: Morgan-Kaufmann.

Falkenhainer, B. (1988). *Learning from physical analogies: A study of analogy and the explanation process* (Tech. Rep. No. UIUCDCS-R-88-1479). Urbana: University of Illinois, Department of Computer Science.

Falkenhainer, B. (1990a). Analogical interpretation in context. *Proceedings of the twelfth annual Conference of the Cognitive Science Society* (pp. 69-76). Cambridge, MA: Hillsdale, NJ: Erlbaum.

Falkenhainer, B. (1990b). A unified approach to explanation and theory formation. In J. Shrager & P. Langley (Eds.), *Computational models of scientific discovery and theory formation* (pp. 157-196). Los Altos, CA: Morgan Kaufmann.

Falkenhainer, B., Forbus, K.D., & Gentner, D. (1986). The structure-mapping engine. *Proceedings of the fifth National Conference on Artificial Intelligence* (pp. 272-277). Philadelphia, PA, Los Altos, CA: Morgan Kaufmann.

Falkenhainer, B., Forbus, K., & Gentner, D. (1989). The structure-mapping engine: Algorithm and examples. *Artificial Intelligence, 41,* 1-63.

Faries, J.M., & Reiser, B.J. (1988). Access and use of previous solutions in a problem-solving situation. *Proceedings of the tenth annual meeting of the Cognitive Science Society* (pp. 433-439). Montreal, Hillsdale, NJ: Erlbaum.

Forbus, K.D., Ferguson, R.W., & Gentner, D. (1994). Incremental structure mapping. *Proceedings of the sixteenth annual Conference of the Cognitive Science Society* (pp. 313-318). Georgia, Hillsdale, NJ: Erlbaum.

Forbus, K., & Gentner, D. (1986). Learning physical domains: Towards a theoretical framework. In R.S. Michalski, J.G. Carbonell, & T.M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach* (Vol. 2, pp. 311-348). Los Altos, CA: Morgan Kaufmann.

Forbus, K., & Gentner, D. (1989). Structural evaluation of analogies: What counts? *Proceedings of the eleventh annual Conference of the Cognitive Science Society* (pp. 341-348). Ann Arbor, MI, Hillsdale, NJ: Erlbaum.

Forbus, K.D., & Gentner, D. (1991). Similarity-based cognitive architecture. Sigart Bulletin, Vol. 2, No. 4, pp. 66-69.

Forbus, K.D., & Oblinger, D. (1990). Making SME greedy and pragmatic. *Proceedings of the twelfth annual Conference of the Cognitive Science Society* (pp. 61-68). Cambridge, MA, Hillsdale, NJ: Erlbaum.

Gentner, D. (1981). Some interesting differences between nouns and verbs. *Cognition and Brain Theory, 4*(2), 161-178.

Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science, 7,* 155-170.

Gentner, D. (1988). Metaphor as structure mapping: The relational shift. *Child Development, 59,* 47-59.

Gentner, D. (1989a). Finding the needle: Accessing and reasoning from prior cases. *Proceedings: Case-Based Reasoning Workshop,* Defense Advanced Research Projects Agency, Information Science and Technology Office (pp. 137-143).

Gentner, D. (1989b). The mechanisms of analogical learning. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 199-241). New York: Cambridge University Press.

Gentner, D., & Clement, C. (1988) Evidence for relational selectivity in the interpretation of analogy and metaphor. In G.H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 22, pp. 307-358).

Gentner, D., & France, I.M. (1988). The verb mutability effect: Studies of the combinatorial semantics of nouns and verbs. In S.L. Small, G.W. Cottrell, & M.K. Tanenhaus (Eds.), *Lexical ambiguity resolution: Perspectives from psycholinguistics, neuropsychology, and artificial intelligence* (pp. 343-382). San Mateo, CA: Morgan Kaufmann.

Gentner, D., & Landers, R. (1985). Analogical reminding: A good match is hard to find. *Proceedings of the International Conference on Cybernetics and Society* (pp. 607-613). Tucson, AZ.

Gentner, D., & Markman, A.B. (1993). Analogy: Watershed or Waterloo: Structural alignment and the development of connectionist models of cognition. In S.J. Hanson, J.D. Cowan, & C.L. Giles (Eds.), *Advances in neural information processing systems 5* (pp. 855-862). San Mateo, CA: Morgan Kaufmann.

Gentner, D., & Markman, A.B. (1994a). Similarity is like analogy. In C. Cacciari (Ed.), *Proceedings of the Workshop on Similarity at the University of San Marino.* Milan, Italy: Bompiani.

Gentner, D., & Markman, A.B. (1994b). Structural alignment in comparison: No difference without similarity. *Psychological Science, 5*(3), 153-158.

Gentner, D., & Ratterman, M.J. (1991). Language and the career of similarity. In S.A. Gelman & J.P. Brynes (Eds.), *Perspectives on language and thought: Interrelations in development* (pp. 225-277).

Gentner, D., Rattermann, M.J., & Forbus, K.D. (1993). The roles of similarity in transfer: Separating retrieval from inferential soundness: *Cognitive Psychology, 25,* 524-575.

Gentner, D., Rattermann, M.J., Markman, A.B., & Kotovsky, L. (1995). Two forces in the development of relational similarity. In G. Halford & T. Simon (Eds.), *Developing cognitive competence: New approaches to process modeling* (pp. 263-313). Hillsdale, NJ: Erlbaum.

Gick, M.L., & Holyoak, K.J. (1980). Analogical problem solving. *Cognitive Psychology, 12,* 306-355.

Gick, M.L., & Holyoak, K.J. (1983). Schema induction and analogical transfer. *Cognitive Psychology, 15,* 1-38.

Gillund, G., & Shiffrin, R.M. (1984). A retrieval model for both recognition and recall. *Psychology Review, 91,* 1-67.

Glucksberg, S., & Keysar, B. (1990). Understanding metaphorical comparisons: Beyond similarity. *Psychological Review, 97,* 3-18.

Goldstein, E., Kedar, S., & Bareiss, R. (1993). Easing the creation of a multi-purpose case library. *Proceedings of the AAAI-93 Workshop on Case-Based Reasoning* (pp. 12-18). Washington, DC, Menlo Park, CA: AAAI Press.

Goldstone, R.L., & Medin, D.L. (1994a). Similarity, interactive-activation and mapping: In K.J. Holyoak & J.A. Barnden (Eds.), *Advances in connectionist and neural computation theory, Vol. 2: Connectionist approaches to analogy, metaphor, and case-based reasoning.* Norwood, NJ: Ablex.

Goldstone, R.L., & Medin, D.L. (1994b). Time course of comparison. *Journal of Experimental Psychology: Learning, Memory and Cognition, 20*(1), 29-50.

Goldstone, R.L., Medin, D., & Gentner, D. (1991). Relational similarity and the nonindependence of features in similarity judgments. *Cognitive Psychology, 23,* 222-264.

Halford, G.S. (1992). Analogical reasoning and conceptual complexity in cognitive development. *Human Development.*

Hammond, K.J. (1986). CHEF: A model of case-based planning. *Proceedings of the fifth*

National Conference on Artificial Intelligence (pp. 267-271). Philadelphia, PA, Los Altos, CA: Morgan Kaufmann.

Hammond, K. (1989). On functionally motivated vocabularies: An apologia. *Proceedings: Case-Based Reasoning Workshop* (pp. 52-56). San Mateo, CA: Morgan Kaufmann.

Hinton, G.E., & Anderson, J.A. (1989). *Parallel models of associative memory.* Hillsdale, NJ: Erlbaum.

Hintzman, D.L. (1984). MINERVA 2: A simulation model of human memory. *Behavior Research Methods, Instruments, & Computers, 16,* 96-101.

Hintzman, D. (1986). 'Schema abstraction' in a multiple-trace memory model. *Psychological Review, 93,* 411-428.

Hintzman, D.L. (1988). Judgments of frequency and recognition memory in a multiple-trace memory model. *Psychological Review, 95,* 528-551.

Holyoak, K.J., & Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory & Cognition, 15,* 32-340.

Holyoak, K.J., Novick, L.R., & Melz, E. (1994). Component processes in analogical transfer: Mapping, pattern completion, and adaptation. In K.J. Holyoak & J.A. Barnden (Eds.), *Advances in connectionist and neural computation theory, Vol. 2: Connectionist approaches to analogy, metaphor, and case-based reasoning.* Norwood, NJ: Ablex.

Holyoak, K.J., & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science, 13,* 295-355.

Humphreys, M.S., Bain, J.D., & Pike, R. (1989). Different ways to cue a coherent memory system: A theory for episodic, semantic, and procedural tasks. *Psychological Review, 2,* 208-233.

Kass, A. (1986). Modifying explanations to understand stories. *Proceedings of the eighth annual Conference of the Cognitive Science Society* (pp. 691-696). Amherst, MA.

Kass, A. (1989). Strategies for adapting explanations. *Proceedings: Case-Based Reasoning Workshop* (pp. 119-123). San Mateo, CA: Morgan Kaufmann.

Keane, M. (1987). On retrieving analogues when solving problems. *Quarterly Journal of Experimental Psychology, 39A,* 29-41.

Keane, M. (1988a). Analogical mechanisms. *Artificial Intelligence Review, 2,* 229-250.

Keane, M.T (1988b). *Analogical problem solving.* Chichester: Ellis Horwood (New York: Wiley).

Keane, M.T. (1988c). *Incremental analogising: Theory & model* (Tech. Rep. No. 38). Milton Keynes, England: The Open University, Human Cognition Research Laboratory.

Keane, M.T. (1990). Incremental analogising: Theory and model. In K. Gilhooly et al. (Eds.), *Lines of thinking.* Chichester: Wiley.

Keane, M.T. (1991, August). Similarity and ordering constraints. *Proceedings of the meeting of the Cognitive Science Society.*

Keane, M., & Brayshaw, M. (1988). The incremental analogy machine: A computational model of analogy. In D. Sleeman (Ed.), *Third European working session on machine learning* (pp. 53-62). San Mateo, CA: Morgan Kaufmann.

Keane. M.T., Ledgeway, T., & Duff, S. (1991, August). Constraints on analogical mapping: The effects of similarity and order. *Proceedings of the thirteenth annual Conference of the Cognitive Science Society* (pp. 275-280). Chicago, Hillsdale, NJ: Erlbaum.

Keil, F.C. (1989). *Concepts, kinds, and cognitive development.* Cambridge, MA: MIT Press.

Kettler, B.P., Hendler, J.A., & Anderson, W.A. (1992). Massively parallel support for a case-based planning system. *Proceedings of the NASA Space Applications and Operations Research Workshop.* Houston, TX.

King, J., & Bareiss, R. (1989). Similarity assessment in case-based reasoning. *Proceedings: Case-Based Reasoning Workshop* (pp. 67-71). San Mateo: Morgan Kaufmann.

Kolodner, J.L. (1984). *Retrieval and organization structures in conceptual memory: A computer model.* Hillsdale, NJ: Erlbaum.

Kolodner, J.L. (1988). *Proceedings of the first Case-Based Reasoning Workshop.* Los Altos, CA: Morgan Kaufmann.

Kolodner, J.L. (1989). Judging which is the "best" case for a case-based reasoner. *Proceeedings: Case-Based Reasoning Workshop* (pp. 77-81). San Mateo, CA: Morgan Kaufmann.

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

Kotovsky, L., & Gentner, D. (1990). Pack light: You will go farther. *Proceedings of the second Midwest Artificial Intelligence and Cognitive Science Society Conference.* Carbondale, IL.

Lange, T.E. & Wharton, C.M. (1992). REMIND: Integrating language understanding and episodic memory retrieval in a connectionist network. *Proceedings of the fourteenth annual Conference of the Cognitive Science Society* (pp. 576-581). Hillsdale, NJ: Erlbaum.

Lange, T.E. & Wharton, C.M. (1993). Dynamic memories: Analysis of an integrated comprehension and episodic memory retrieval model. *Proceedings of the thirteenth International Joint Conference on Artificial Intelligence* (pp. 208-213). Chambery, San Mateo, CA: Morgan Kaufmann.

Markman, A.G., & Gentner, D. (1990). Analogical mapping during similarity judgments. *Proceedings of the twelfth annual Conference of the Cognitive Science Society* (pp. 38-44). Cambridge, MA, Hillsdale, NJ: Erlbaum.

Markman, A.G., & Gentner, D. (1993a). Splitting the differences: A structural alignment view of similarity. *Journal of Memory and Language.*

Markman, A.B., & Gentner, D. (1993b). Structural alignment during similarity comparisons. *Cognitive Psychology, 25,* 431-467.

Martin, C. (1989). Indexing using complex features. *Proceedings of the AAAI-93 Workshop on Case-Based Reasoning* (pp. 26-30). Washington, DC, Menlo Park, CA: AAAI Press.

Medin, D.L., Goldstone, R.L., & Gentner, D. (1993). Respects for similarity. *Psychological Review, 100*(2), 254-278.

Medin, D., & Ortony, A. (1989). Psychological essentialism. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 179-195). New York: Cambridge University Press.

Medin, D.L., & Ross, B.H. (1989). The specific character of abstract thought: Categorization, problem-solving, and induction. In R.J. Sternberg (Ed.), *Advances in the psychology of human intelligence* (Vol. 5, pp. 189-223). Hillsdale, NJ: Erlbaum.

Medin, D.L., & Schaffer, M.M. (1978). Context theory of classification learning. *Psychological Review, 85,* 207-238.

Metcalfe, J. (1993). Novelty monitoring, metacognition, and control in a composite holographic associative recall model: Implications for Korsakoff amnesia. *Psychological Review, 100*(1), 3-22.

Miller, G., Fellbaum, C., Kegl, J., & Miller, K. (1988). WordNet: An electronic lexical reference system based on theories of lexical memory. *Revue Quebecoise Linguistique, 16,* 181-213.

Minsky, M. (1981). A framework for representing knowledge. In J. Haugland (Ed.), *Mind design* (pp. 95-128). Cambridge, MA: MIT Press.

Murphy, G.L., Medin, D.L. (1985). The role of theories in conceptual coherence. *Psychological Review, 92,* 289-312.

Novick, L.R. (1988a). Analogical transfer: Processes and individual differences: In D.H. Helman (Ed.), *Analogical reasoning: Perspectives of artificial intelligence, cognitive science, and philosophy* (pp. 125-145). Dordrecht, The Netherlands: Kluwer.

Novick, L.R. (1988b). Analogical transfer, problem similarity, and expertise. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14,* 510-520.

Pazzani, M. (1989). Indexing strategies for goal-specific retrieval of cases. *Proceedings: Case-Based Reasoning Workshop* (pp. 31-35). San Mateo, CA: Morgan Kaufmann.

Porter, B.W. (1989). Similarity assessment: Computation vs. representation. *Proceedings: Case-Based Reasoning Workshop* (pp. 82-84). San Mateo, CA: Morgan Kaufmann.

Ratcliff, R. (1990). Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological Review, 97,* 285-308.

Ratcliff, R., & McKoon, G. (1989). Similarity information versus relational information: Differences in the time course of retrieval. *Cognitive Psychology, 21,* 139-155.

Rattermann, M.J., & Gentner, D. (1987). Analogy and similarity: Determinants of accessibility and inferential soundness. *Proceedings of the ninth annual meeting of the Cognitive Science Society* (pp. 23-34). Seattle, WA.

Read, S.J., & Cesa, I.L. (1991). This reminds me of the time when...: Expectation failures in reminding and explanation. *Journal of Experimental Social Psychology, 27,* 1-25.

Reder, L.M. (1988). Strategic control of retrieval strategies. *The Psychology of Learning and Motivation, 22,* 227-259.

Reed, S.K. (1987). A structure-mapping model for word problems. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 13,* 124-139.

Reed, S.K., Ackinclose, C.C., & Voss, A.A. (1990). Selecting analogous problems: Similarity versus inclusiveness. *Memory & Cognition, 18*(1), 83-98.

Reed, S.K., Ernst, G.W., & Banerji, R. (1974). The role of analogy in transfer between similar problem states. *Cognitive Psychology, 6,* 436-450.

Reeves, L.M., & Weisberg, R.W. (1994). The role of content and abstract information in analogical transfer. *Psychological Bulletin, 115*(3), 381-400.

Riesbeck, C.K., & Schank, R.C. (1989). *Inside case-based reasoning.* Hillsdale, NJ: Erlbaum.

Rips, L.J. (1989). Similarity, typicality, and categorization. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 21-59). New York: Cambridge University Press.

Ross, B.H. (1984). Remindings and their effects in learning a cognitive skill. *Cognitive Psychology, 16,* 371-416.

Ross, B.H. (1987). This is like that: The use of earlier problems and the separation of similarity effects. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 13,* 629-639.

Ross, B.H. (1989). Remindings in learning and instruction. In S. Vosniadou & A. Ortony (Eds.), *Similarity and analogical reasoning* (pp. 438-469). New York: Cambridge University Press.

Schank, R. (1982). *Dynamic memory.* New York: Cambridge University Press.

Schumacher, R., & Gentner, D. (1987, May). *Similarity-based remindings: The effects of similarity and interitem distance.* Paper presented at the Midwestern Psychological Association, Chicago, IL.

Seifert, C.M., McKoon, G., Abelson, R.P., & Ratcliff, R. (1986). Memory connection between thematically similar episodes. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 12,* 220-231.

Skorstad, J., Falkenhainer, B., & Gentner, D. (1987). Analogical processing: A simulation and empirical corroboration. *Proceedings of the sixth National Conference on Artificial Intelligence* (pp. 322-326). Seattle, WA: Los Altos, CA: Morgan Kaufmann.

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* (pp. 419-425). Montreal, Canada.

Smolensky, P. (1988). On the proper treatment of connectionism. *Behavior and Brain Sciences, 11,* 1-74.

Spellman, B.A., & Holyoak, K.J. (1992). If Saddam is Hitler then who is George Bush? Analogical mapping between systems of social roles. *Journal of Personality and Social Psychology, 62*(9), 913-933.

Stanfill, C., & Waltz, D. (1986). Toward memory-based reasoning. *Transaction of ACM, 29*(12), 1213-1228.

Thagard, P., & Holyoak, K.J. (1989). Why indexing is the wrong way to think about analog retrieval. *Proceedings: Case-Based Reasoning Workshop* (pp. 36-40). San Mateo, CA: Morgan Kaufmann.

Thagard, P., Holyoak, K.J., Nelson, G., & Gochfeld, D. (1990). Analog retrieval by constraint satisfaction. *Artificial Intelligence, 46,* 259-310.

Van Lehn, K. (1989). *Mind bugs: The origins of procedural misconceptions.* Cambridge, MA: MIT Press.

Waltz, D. (1989, May). Panel discussion on "indexing algorithms." *Proceedings: Case-Based Reasoning Workshop* (pp. 25-44). San Mateo, CA: Morgan Kaufmann.

Wharton, C.M., Holyoak, K.J., Downing, P.E., Lange, T.E., & Wickens, T.D. (1991). Retrieval competition in memory for analogies. *Proceedings of the thirteenth annual Conference of the Cognitive Science Society* (pp. 528-533). Hillsdale, NJ: Erlbaum.

Wharton, C.M., Holyoak, K.J., Downing, P.E., Lange, T.E., & Wickens, T.D. (1992). The story with reminding: Memory retrieval is influenced by analogical similarity. *Proceedings of the fourteenth annual Conference of the Cognitive Science Society* (pp. 588-593). Hillsdale, NJ: Erlbaum.

Wharton, C.M., Holyoak, K.J., Downing, P.E., Lange, T.E., Wickens, T.D., & Melz, E.R. (1994). Below the surface: Analogical similarity and retrieval competition in reminding. *Cognitive Psychology, 26,* 64-101.

Winston, P.H. (1982). Learning new principles from precedents and exercises. *Artificial Intelligence, 19,* 321-350.