MAC/FAG
A Model of Similarity-based Retrieval

KENNETH D. FORBUS,
DEDRE CENTNER,
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) reminding» 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 Bursiein, Gregg Collins, Ron Ferguson, Brian Falkenhainer, Rob Goldstone, An Markman, Doug Medin, and Mary Jo Radermann 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 ю Kenneth D. Forbus, Northwestern University, The Institute for the Learning Sciences, 1890 Maple Avenue, Evanslon. 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 (Centner, 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 dam sets) and that MAC/FAC's oerformance fits the psyhological eviceace bener 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 (Faikenhainer, 1990a, b; Holyoak, Novick, & Melz, 1994; Kass, 1986, 1989), and *abstracting* the common structure from base and target (Gick & Holyoak, 1983; Skorstad, Centner, & 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 (Centner, 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 (Centner, 1983; Centner & Markman, 1993, 1994a, 1994b; Goldstone & Medin, 1994a, 1994b; Goldstone, Medin, & Centner, 1991; Markman & Centner, 1990, 1993a, 1993b; Medin, Goldstone, & Centner, 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-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 Centner (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 (Centner, 1988; Centner & Clement, 1988). Centner, 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 (Centner & Landers, 1985; Centner, Rattermann, & Forbus, 1993; Rattermann & Centner, 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 & Centner, 1990, 1993b) and determine common and distinctive features (Centner & Markman, 1994a, b; Markman & Centner, 1993a).

[1] We define the *order* of an item in a representation as follows: Objects and constants are order 0; the order of a statement is I 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; Centner & Markman, 1994a, b; Goldstone et al., 1991; Holyoak et al., 1994; Keane, 1988a, 1988b; Markman & Centner, 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 (Centner & Landers, 1985; Centner, Rattermann, & Forbus, 1993; Rattermann & Centner, 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 fronr 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 (Paries & 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 tradeoffs 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. Ho·..: . , ·': models also Ivnicallv пгмипк» a hiphlv ind>Yi»rl mpmnrv in whirh ;h? vnrs 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' to 10') 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; Hint/man, 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 nonslructured knowledge representations and relatively simple match processes and hence do not allow for structural matching and inference. Such models also (end to use a unitary notion of similarity, an assumption that is called into question by the dissociation described earlier (see also Centner & 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. THK 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 & Uareiss, 1989;Waltz, 1989). MAC/FAC uses this strategy. The disassociation noted previously can be understood in terms of the interactione of its

Figure I illustrates the components of the MAC/FAC model. The inputs are a pool of memory items and л *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.

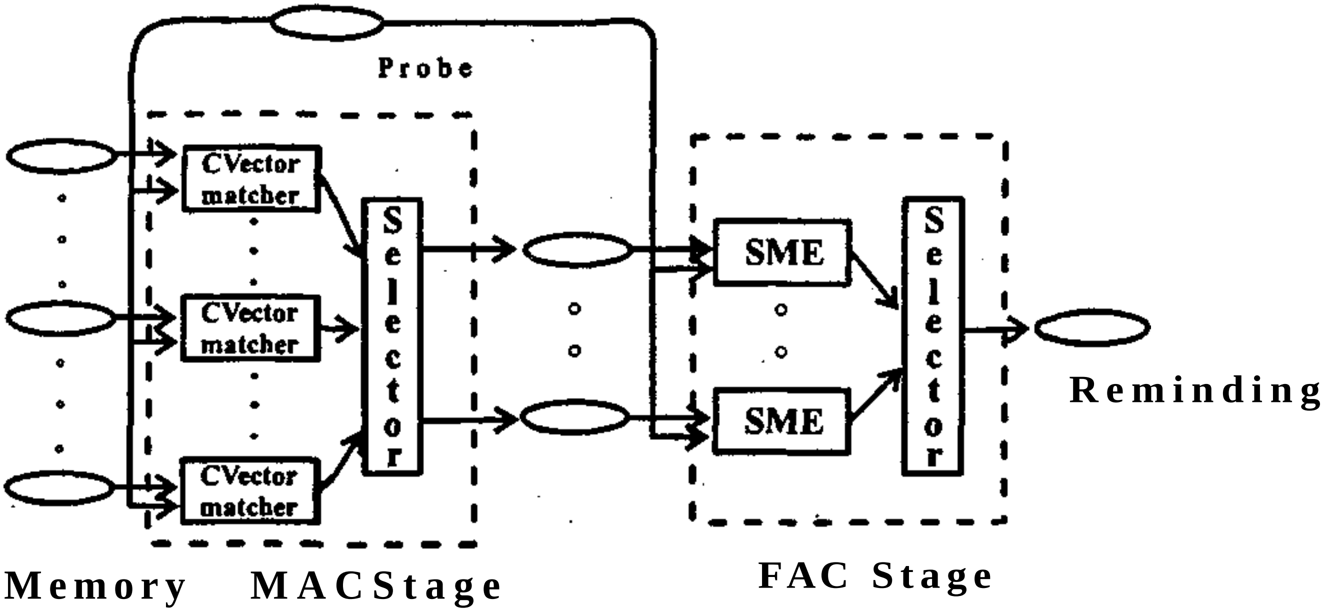
We begin by describing the FAC stage. In doing so, we also describe the computational framework which underlies MAC and FAC, including our conventions for representation and the information about the SME algorithm that is required to fully understand MAC/FAC.

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, & Centner, 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 япН ι *,ro*t*



Flgur· 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 arc not included in the interpretation. We refer to this preference for connected systems of common predicates as the *systemaiicity* 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.[1]

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

[1] 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 *i.* v..: : ;. . ..,:ignable) Structure denends nn the reasoner's state of knowlege about the target

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 Bj and target item Tj when either

1. Bj and Tj are both statements whose functors are sufficiently alike (typically identical, but see below), or
2. Bj and TJ 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 Event! 7 Event31)
T1: (CAUSE Events 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 Events match, and also that Eventsi and Eventea 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)

82 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 identically.* 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* (Centner, 1989; Centner & 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 ЛЧо-опе 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 RoboU, 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 ei 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, & Centner, 1994; Forbus & Oblinger. 1990).

' We thank AxihurMarkman for the drawings of !x;urc 2 anUiheco

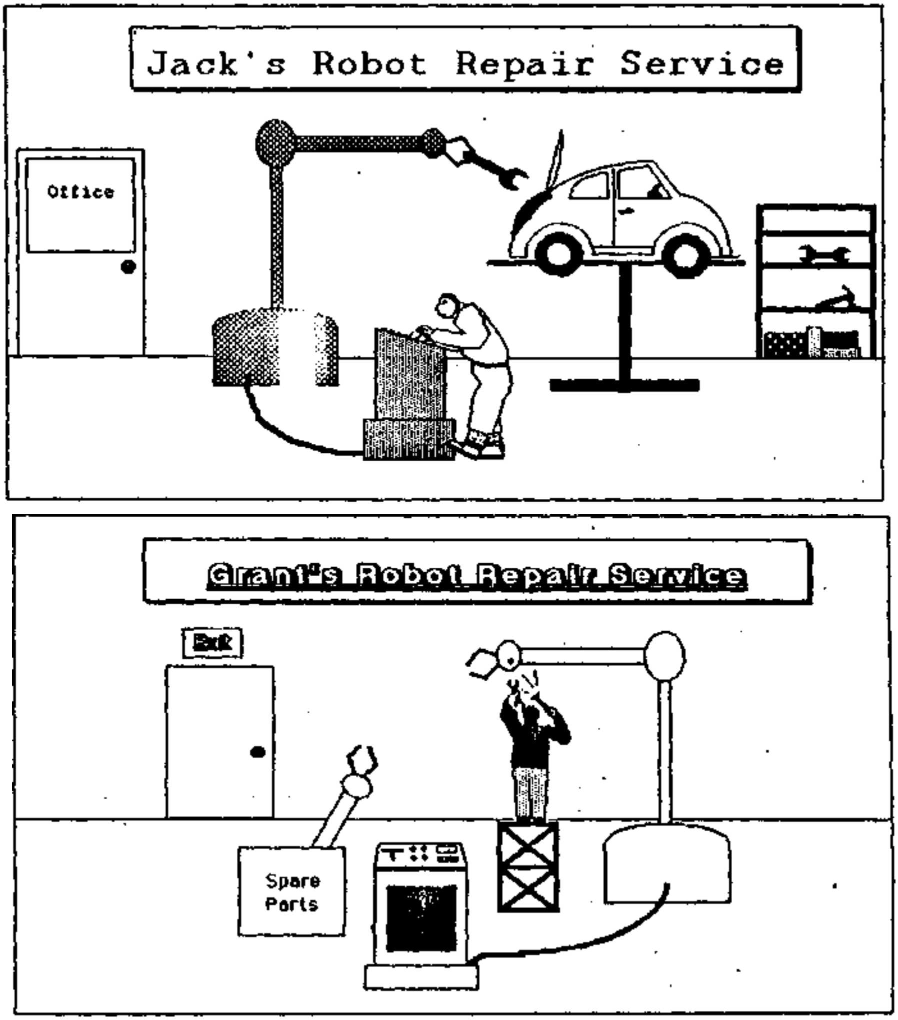
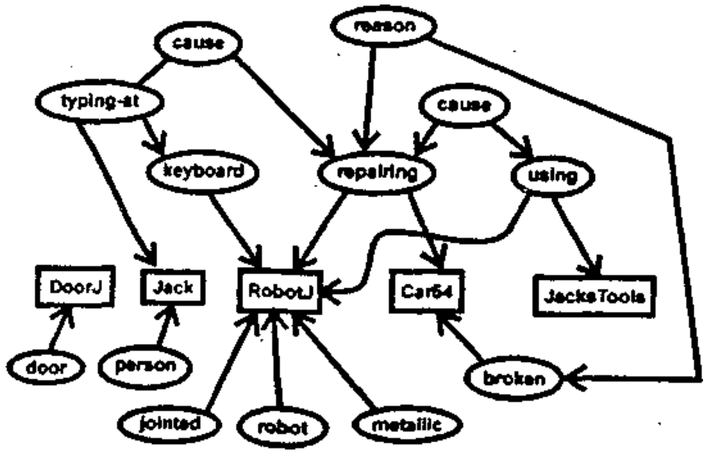


Figure 2. Two simple situations.

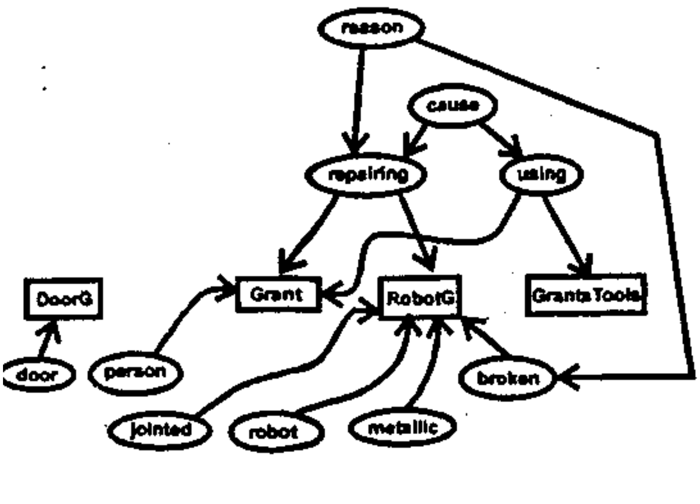
The third phase is structural evaluation. For simplicity, we describe this stage as conceptually distinct from the previous stage, although it is actually interleaved with building interpretations, because its results guide the greedy merge algorithm. To capture human preferences, the structural evaluation 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 ;,,:-;.ierence. 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)

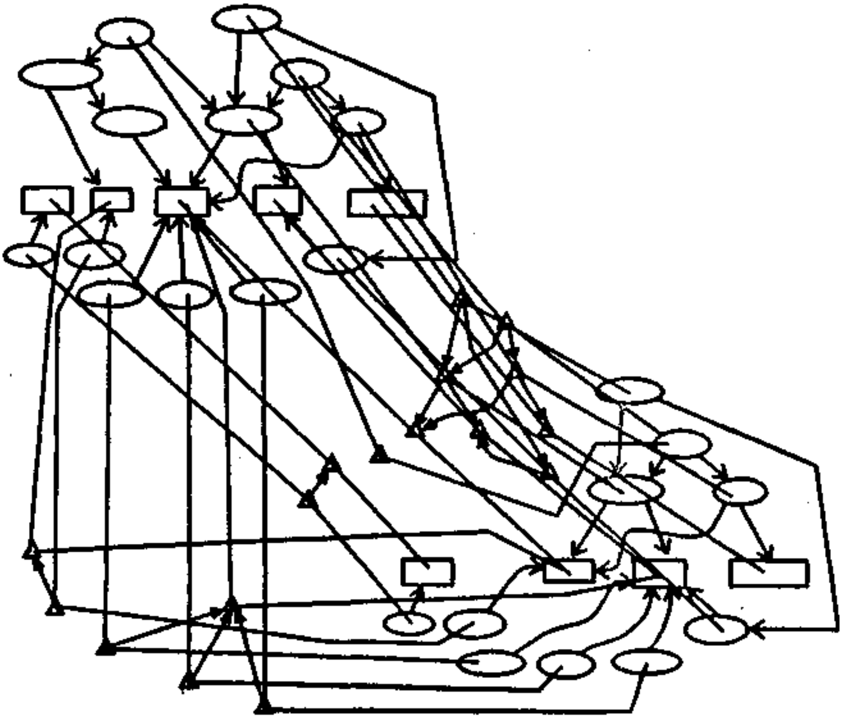


Figure 4. A match hypothesis forest. This picture illustrates the match hypotheses generated for a pair of simple descriptions. Match hypotheses are shown as triangles. Dashed lines indicate the base and target items each match hypothesis places in correspondence. The solid arrows leaving a match hypothesis 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.[9] That is, if W (MH,) is the score associated with a match hypothesis MH_b MH_2 is a match hypothesis that applies to one of MHi's arguments, and δ is the trickle-down factor, then W (MH_2) is incremented as follows:

W (MHz) - max {W (MHz) + 6W(MH,);1.0}

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

[1] The systematicity preference could have been implemented by differentially weighting matches at different levels. This method would seem to require a co .;-!';r,;опа!!y implausible "bird's-eye" view of the representations. In a comparison of the ι. . . . . ,c irickle-
rlnurn rtiMhrwi arrnunl^rt fnr human cmmHn» rofinnr K»it»r tho« II·<*I^П.I ..r..;.»!.!,· /:;r^ti«i nr *»**

```
GM1: 10 correspondences, SES =4.66
  Object mappings:
   DOORG <-> DOORJ
   ROBOTG <-> CARS4
   GRANT <-> ROBOTJ
   HANDTOOLSG <-> HANDTOOLSJ
  No candidate inferences.
```

FIgur· 5. Global interpretations (or 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 fur 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 (Centner & Landers, 1985; Rattermann & Centner, 1987). Like humans, SME rated analogical matches higher than surface matches (Skorstad, Falkenhainer, & Centner, 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 (Centner, Rattermann, & Forbus, 1993).

Because reirit' ., occur frequently, components in model of retrieval and should typically be better than O(log(n)) on data-parallel machines.' The generation of global interpretations is roughly O(log(n)> on a serial machine, using the greedy merge algorithm of Forbus and Oblinger (1990),' 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.' 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 & Centner, 1986; Centner, 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.

' 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.

' 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.

' In our current implementation, SME is run sequentially or ' didate item in turn,

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 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 oцr original version of MAC/FAC did (Centner, 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.' 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 л-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).

• ARCS is based on Holyoak and Thagard's (1989) ACME, an a::·-·'-*?.y* matcher which uses a localise connettionist network similar to SME's match hypothesis . . . ., .. .:-.uucl a single interoreiation of a comoarison via constraint satisfaction.

SoUr System: Structured representation

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

Sotor 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 1
(GREATER ·. 1)
(MASS . 2)
(OBJECTS . 2)
(OPPOSITE-SIGN . 1)
(REVOLVE-AROUND . I)
```

FIgur- 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

" We normalize corner vectors to unit vectors, both to reduce the sensitivity to overall size of the descriptions anJ .;.\:;.ise we assume that psychologically plausible implementation substrate for MAC/FAC !.- : ;-nral «м.1«™\ ««Ч i

TABLE 1
Types of Stories Used in the "Korlo 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 (Centner, Rattermann, & Forbus, 1993; Rattermann & Centner, 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 con-

TABLE 2
Proportion of Reminding; for Different Motch Types: Human Participants

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

Note. LS=literal similarity: SF=surf**ace**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."

As discussed above, this retrievability order differs strikingly from the soundness ordering. When subjects were asked to rate howдоиш/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 & Centner, 1989; Centner 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.

" LS and SF did not differ significantly in retrievability. In Experiment 2, AN and FOR did not differ significantly, although in Experiment I, AN matches were better retrieved than

```
(FOLLOW
  «PROMISE MAN1 KARLA
    (NOT (ATTACK MAN1 KARLA) ) )
  (ATTACK HANI DEER))
(CAUSE
 (EQUALS (HAPPINESS MAN1) HIGH)
 (PROMISE MAN1 KARLA
   (NOT (ATTACK HANI KARLA))))
(CAUSE
 (OBTAIN MAN1 FEATHERS)
 (EQUALS (HAPPINESS MAN!) HIGH»
(FOLLOW
 (OFFER KARLA FEATHERS HANI)
 (OBTAIN MAN! FEATHERS))
(CAUSE
 (REALIZE KARLA
     (DESIRE MAN1 FEATHERS))
 (OFFER KARLA FEATHERS HANI))
(FOLLOW
 (EQUALS
  (SUCCESS
     (ATTACK MAN1 KARLA)) F)
 (REALIZE KARLA
   (DESIRE MAN1 FEATHERS»)
(CAUSE
 (NOT (USED-FOR
         FEATHERS CROSS-BOW))
 (EQUALS (SUCCESS
           ATTACK HANI KARLA))
                F»
(FOLLOW
 (ATTACK MAN1 KARLA)
 (EQUALS (SUCCESS
     (ATTACK MAN1 KARLA))
                F»

(FOLLOW
  (SEE KARLA MAN!)
  (ATTACK MAN1 KARLA))
(HAPPEN (SEE KARLA HANI»
(LIVES KARLA LOCI)
(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 MAN!)
(HUMAN MAN!)
(MALE MAN1)
(PREDATORY KARLA)
(BLACK KARLA)
(POWERFUL KARLA)
(LARGE KARLA)
(HAWK KARLA)
```

***H'jurm* 7. A representation from th* Karla** 'he 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 ail 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 memo; - r:..rc$e:.:auons.

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) ordinallv match th* rwnitc fnr^

TABLES
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 m?'ches 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

It is useful to compare MAC/FAC's performance with that of Thagard et al.'s (1990) ARCS model dfsimilarity-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 r SF > AN > 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

Given a pool of memory items 11 „In and a probe P:

1. **For each item li, include it in a matching network if there are any predicates in li that are semanlically 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

6.1 Theoretical Trade-Oils

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 ! ·; ,rd, cow]-HEAVIER [«iraffe. donkevl