Where Do Relations Come From?

Michael Gasser and Eliana Colunga
Computer Science Department, Cognitive Science Program
Indiana University

Abstract

Relational knowledge is a hallmark of human cognition and the subject of a vast body of research. In this paper we argue that existing accounts of relations are inadequate because they have little to say about how relations arise in the first place and because they tend to be limited to particular sorts of relational tasks. We present a new approach to the learning and representation of relations, an approach that makes use of what we call **micro-relation units** (MRUs). Each MRU represents a relation between *features* of different objects rather than between objects themselves. We show how this approach offers an account of the grounding of relations, and we describe a neural-network implementation of the MRU framework and show how it enables a variety of relational tasks to be performed by the same system.

Introduction

In this paper, we present a new approach to the learning and representation of relations in neural networks, an approach that makes use of what we call **micro-relation units** (MRUs). Each MRU represents a relation between *features* of different objects rather than between objects themselves. We show how this approach, unlike others, offers an account of the grounding of relations and a means by which a variety of relational tasks can be performed by the same system.

The paper is organized as follows. First we discuss the problem of representing and learning relations within the context of "grounded" models of language. We consider a number of alternative ways in which relations have been handled in neural networks and explain why each of these is inadequate. Next we introduce the MRU framework, emphasizing how it overcomes deficiencies with other frameworks. Next, we present simulations of the learning of several simple relations to illustrate the MRU approach. Finally, we consider some behavioral predictions that MRUs lead to.

Relational Knowledge

One of the hallmarks of human reasoning and linguistic behavior is the use of relational knowledge. Relations are fundamental to any account of linguistic semantics: the universal distinction between nouns and verbs may be a reflection of a universal cognitive distinction between objects and relations(Langacker, 1987). The ability to reason in terms of relations appears to distinguish people from most or all other animals. (Pearce, 1994). And, beginning at the age of 4, people's judgments of the similarity of two items tends to be based on relational rather than purely featural similarity(Gentner, Rattermann, Markman, & Kotovsky, 1995). Any account of relational knowledge must deal with the representation of knowledge as well as with the ways in which it is used in performing particular cognitive tasks. In what follows, we consider these in turn.

Relational Representation

Informally at least, a (binary, non-reflexive) relation associates two distinct objects with one another. Thus any account of relations presupposes an account of objects. Most accounts take the objects associated by a relation to be atomic with respect to the relation; that is, the objects in a relation are in a sense prior to it. Below we will offer an alternative account in which relations are built out of object features rather than objects. For the moment, however, we will represent objects as nodes in a network; we will refer to these as **object units (OUs)**. The simplest association between objects would take the form of a line connecting the object nodes (Figure 1a).

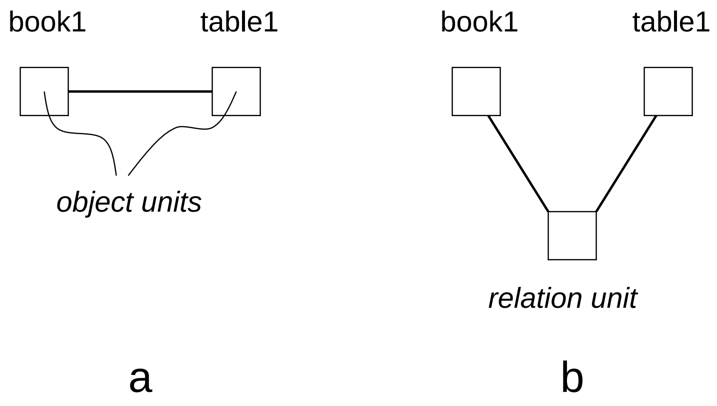

Figure 1. Two insufficient ways of representing relations between objects: a simple connection between object units (a), two associated objects connected to a third unit that represents the relation (b).

Relational knowledge, however, goes beyond this sort of simple associative knowledge in that the association between the related elements is made explicit and accessible to the rest of the system (Phillips, Halford, & Wilson, 1995). Among other consequences, this property of relational knowledge permits the representation and learning of associations between relations. For example, a system endowed with relational knowledge could represent the associations required to make the inferences that if X is above Y, then Y is below X, X will land on Y if released, etc.

Explicit reference to the association between two objects requires (at least) a node representing the binary association; we will call this the **relation unit (RU)**. The RU becomes active (accessible, etc.) to the extent that the two OUs which feed into it are active. With bidirectional connections, the RU can also activate the OUs, allowing the system to access the associated objects given the association between them. Figure 1b shows these relationships.

However, it should be immediately obvious that the arrangement shown in Figure 1b falls short of representing what is usually thought of as relational knowledge. A relation is not just two objects: assigning a single node to the association between a book and a table tells us nothing about how the book and table are related. Traditionally conceived, a relation includes two further kinds of information: (a) an element that characterizes the content of the relation and its arity, the **relation term**, and (b) a set of **bindings**, a mapping of the object arguments onto roles in the relation (Halford, Wilson, & Phillips, 1998). The binding of objects to roles is required in order to distinguish, for example, the situation in which a book is above a table from the situation in which a table is above a book. This conventional view of relations characterizes all existing approaches to the representation of relations, whether symbolic or connectionist. Before considering an alternative view, we discuss briefly how these two forms of information are realized in different frameworks.

In symbolic models, the relation term is an explicit symbol such as ABOVE. In distributed connectionist models, such as that of Halford et al. (1994), the relation term takes the form of an activation vector. There are two ways in which the binding information can be implemented. One assigns particular *positions* to the roles and inserts representations of the objects in these positions. This is the approach used in standard predicate-calculus notation: ABOVE(BOOK, TABLE), illustrated in Figure 2.

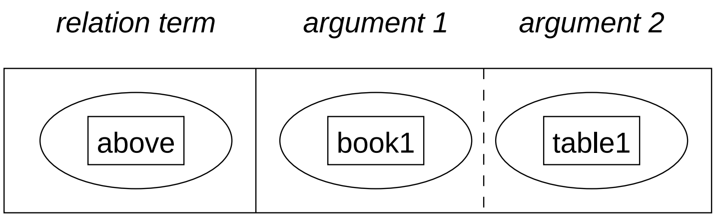

Figure 2. A relation represented using symbolic argument-style representation. The relation term and the arguments are symbols, the bindings are represented by the positions of the arguments.

Within connectionist networks, this approach is used in the model of Halford et al. (Halford et al., 1994). Here the relation term and the related objects, which are all activation vectors, are fed into banks of units which are dedicated to representing the components of the relation. The tensor product of these three vectors (for a binary relation) is computed to complete the binding process. This approach is illustrated in Figure 3.

A second approach to binding involves pairing the objects with role (slot) names. In symbolic role-filler approaches, illustrated in Figure 4, the pairing takes the form of the concatenation of a role-name symbol and an object symbol.

In connectionist slot-filler approaches there are two techniques for implementing the binding. Smolensky's tensor-product framework (1990) and Plate's convolution framework (1995) make use of an approach which is similar to that of Halford et al. (1994). For each role-filler pair, a role-name vector and an object vector are fed into banks of role and filler units respectively, and the tensor product or convolution of these vectors is calculated. The relation representation is the sum of the role-filler products. Note that the relation term may be left out if it is completely specified by the role names; e.g., in place of ABOVE we

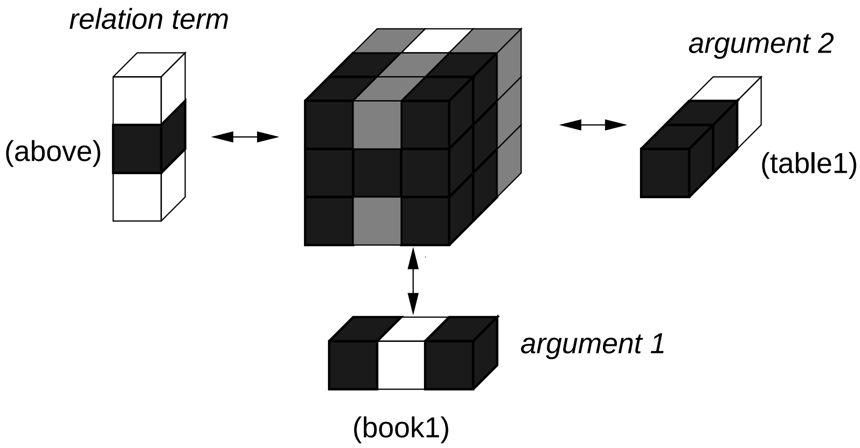

Figure 3. A relation represented using a connectionist argument-style representation. The arguments are fed to dedicated banks of units, and their bindings are represented using the tensor product.

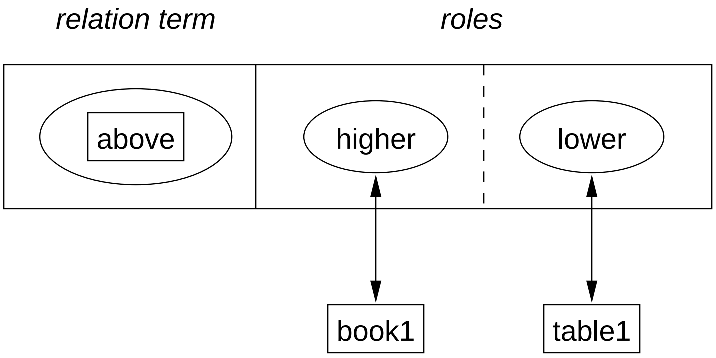

Figure 4. A relation represented using the symbolic explicit role representation. The binding is achieving by concatenating the role-name symbol and the filler object symbol.

have ABOVE-HIGHER and ABOVE-LOWER. This approach is illustrated in Figure 5.

In other connectionist approaches, instead of placing the role and filler in special-purpose banks of units, separate role and filler units are somehow marked as belonging together. That is, in addition to activation, each unit in the network has an associated value which, when it matches the value of another unit, represents a binding between them. In the dynamic binding approach (Hummel & Biederman, 1992; Hummel & Holyoak, 1997; Shastri & Ajjanagadde, 1993; Sporns, Gally, George N. Reeke, & Edelman, 1989) units "fire" at particular times, and units whose firings are synchronized are considered bound.[1] This localist approach is illustrated in Figure 6.

Table 1 summarizes these various approaches to the representation of relational knowledge.

All of these approaches assume that the specification of *how* the objects in a relation

[1]Tesar and Smolensky (1994) have argued that the dynamic binding approach is formally reducible to the tensor product approach.

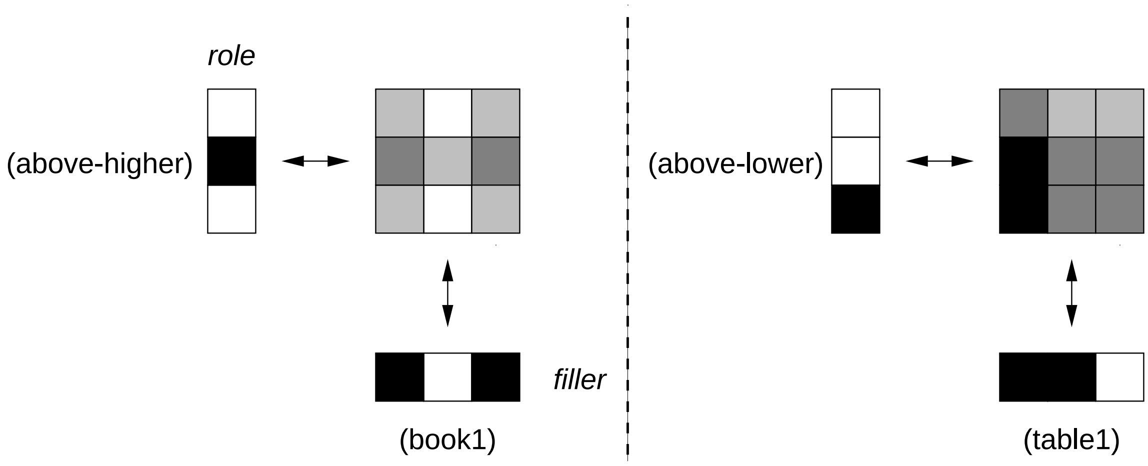

Figure 5. A relation represented using a distributed connectionist explicit role representation. The binding of a role and its filler is computed using the tensor product or convolution.

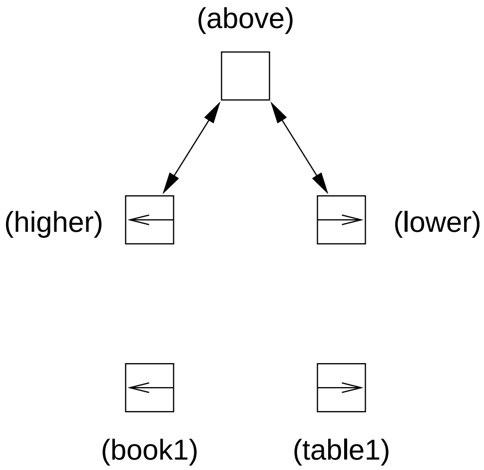

Figure 6. A relation represented using a localist connectionist explicit role representation. Binding is achieved through a value that is shared by the bound role and filler (arrows in the figure).

are related to one another takes the form of an explicit relation term (or explicit role names) together with a mechanism for binding the objects to the roles of the relation. But none of these approaches tells us where the relation term or the roles come from. Relational categories, like object categories, are certainly learned. What process discovers categories like ABOVE, and what sort of substrate does it create symbolic relations from? To consider this question, we need to examine how specific **relation instances** are handled. A relation instance is an explicit association of a particular pair of objects, for example, the spatial relation between a book and a table above which the book is suspended. Just as object categories such BOOK are generalizations over instances of the category, relational categories such as ABOVE are generalizations over instances of the category (Kersten & Billman, 1997).

What alternatives are there for representing relation instances? For simplicity, we

Table 1: Approaches to the representation of relational knowledge.

		Relation term	Bindings
Symbolic	Predicate calculus	Symbol	Symbols in argument positions
	Slot-filler	Symbol	Role symbol + filler symbol
Connectionist	Argument style	Vector	Tensor product of relation and filler vectors
	Distributed, explicit role	(Implicit in bindings)	Sum of tensor product or convolution of role and filler vectors
	Localist	Unit	Role and filler units, synchronized

will consider mainly *spatial* relations in what follows. We could represent a spatial relation instance with an image-like representation which preserves the relative location of the objects. But this would not constitute a *relational* representation because it would provide no means of accessing the relation between the objects (or the objects themselves for that matter); the relation is not made explicit. Alternately, we could represent the relation instance using the abstract categories ABOVE, BOOK, and TABLE. But this approach would offer no insight into how terms such as ABOVE are grounded in perceptual input. We propose a third alternative, one which is relational in the sense that relational associations are directly accessible, but which specifies *how* the objects are related to one another in terms of object features and inter-relation associations rather than abstract symbols.

In our approach, relation categories are built up from relational feature correlations learned on the basis of relation instances, much as object categories are built up from from object feature correlations on the basis of object instances. For our purposes, object instances are cognitive entities (rather than entities in the external world) consisting of values on each of a set of dimensions, such as COLOR and SIZE. Object categories, such as BLOCK, take the form of ranges of values on each dimension, minimally one dimension, but more often patterns of correlations among the values on different dimensions (Rosch, Mervis, Gray, Johnson, & Boyes-Braem, 1976). Object feature correlations begin with the inter-feature connections that are created (or strengthened) with the presentation of an object instance. Figure 7 illustrates object instances, object feature correlations, and object categories.

When two objects are available simultaneously, there is the potential for an explicit connection between features of different objects, that is, a relation instance. With the presentation of multiple relation instances with similar values for the different dimensions, a **relational correlations** is created. Consider first the case of a single dimension. Each of the two objects in a relation instance has a value on that dimension. Multiple relation instances of this sort may lead to a **one-dimensional relational correlation**. For example, if two objects are frequently seen together, one might learn a correlation between the

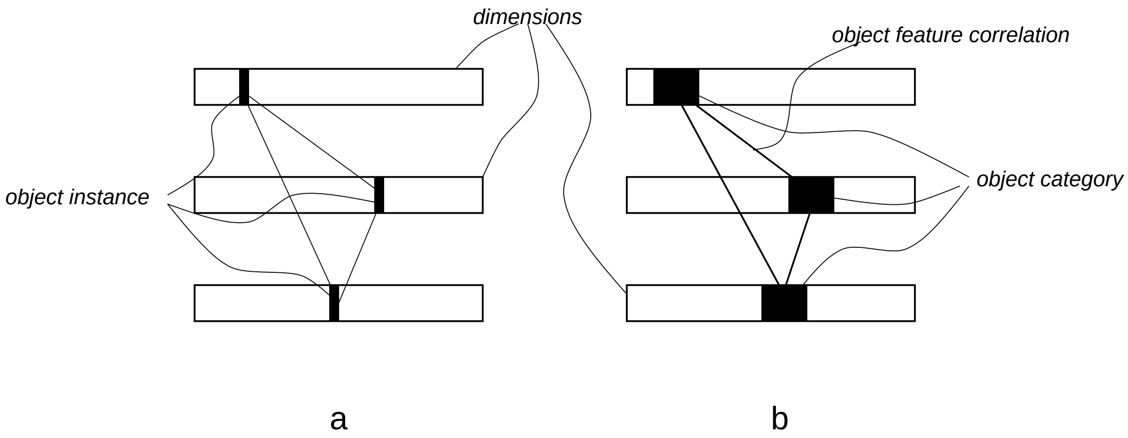

Figure 7. An **object instance** (a) consists of values on each of a set of dimensions. An **object category** (b) consists of *ranges* of values on the different dimensions defined by the correlations (bold lines) found over multiple instances.

sizes of the objects. Figure 8 illustrates a relation instance and a one-dimensional relational correlation.

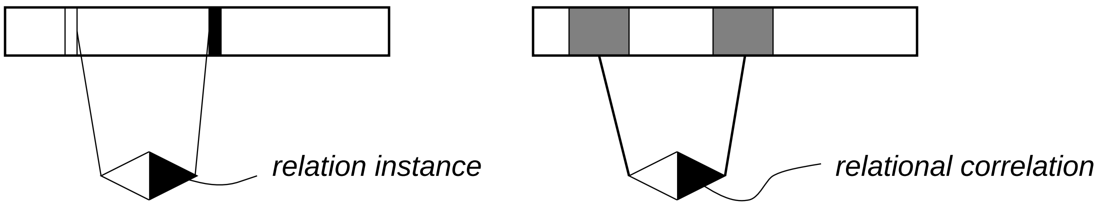

Figure 8. A **relation instance** (a) is two objects presented simultaneously. If the values of the two objects along a particular dimension correlate over several relation instances, a **one-dimensional relational correlation** can be created (b).

Note that the connections representing the relational correlations must be distinguished from those used to represent object feature correlations because relational correlations connect features of *distinct* objects. Below we will argue that relational correlations must be represented not by simple connections, but by separate **micro-relation units (MRUs)**. In the figures these units appear as diamonds.

With experience with a particular dimension, a learner may generalize from narrow regions of values for the two objects to **relative** values across the whole dimension. For example, one may learn to categorize objects as being NEAR or FAR from each other. One way in which this could happen[2] is through the association of more specific **absolute relational correlations** with one another through a **relational category** unit, as shown in Figure 9 for NEAR. The category unit must point to each of the relational correlations rather than to the correlated values; thus each relational correlation must take the form of

[2]In this paper we do not offer an account of how a learner achieves knowledge of relative values across a dimension.

an explicit unit rather than a simple connection. Both the correlations and the category unit are MRUs.

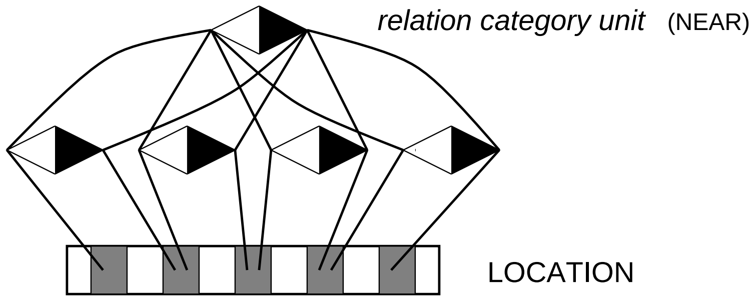

Figure 9. An example of how a relational category unit can be used to represent a relational category (NEAR) by connecting several units representing specific absolute relational correlations along a dimension (LOCATION).

Now consider relations across multiple dimensions. A relation instance in this case starts with a pair of values for each dimension. Given a number of instances, a learner may pick up relational correlations *across* dimensions. There are two possibilities for the sort of knowledge that is involved on a given dimension in a cross-dimensional relational correlation. In one case, the correlation makes reference to a value for one of the objects on the dimension, but not the other. For example, consider the knowledge that if objects X and Y are in vertical contact, X above Y, then object Y has a flat upper surface. This knowledge is represented by a correlation between values on the LOCATION and SHAPE dimensions. For the shape dimension it is only object Y that enters into the correlation. Figure 10a illustrates this possibility. The other possibility is that the values of both objects on the dimension enter into the correlation. This is the case for the LOCATION dimension in the VERTICAL-CONTACT/FLAT-UPPER-SURFACE example above. The locations of both objects are relevant for the correlation. An example in which both objects are relevant for both dimensions is the knowledge that two particular (ranges of) values for the SIZE of two objects correspond to two particular (ranges of) values for the LOUDNESS of the objects. This possibility is illustrated in Figure 10b. Note that the representation of this relational correlation again requires MRUs; simple connections joining pairs of features on different dimensions would not capture the relational nature of the correlation. For example, a connection joining a particular size and a particular loudness value would only represent the tendency for objects of that size to have that loudness.

As with a single dimension, a learner can generalize from absolute values to relative values across one or more of the dimensions. That is, the knowledge about the relationship between SIZE and LOUDNESS could take the more abstract form of the knowledge that relative size, wherever on the size scale, correlates with relative loudness, wherever on the loudness scale.

Given MRUs representing relational correlations, how would such a system handle an input relation instance? A pair of input object instances consist of activated values for each relevant object dimension. For each between-object pair of feature units within a dimension, there is the possibility of an MRU representing a correlation. Such MRUs would

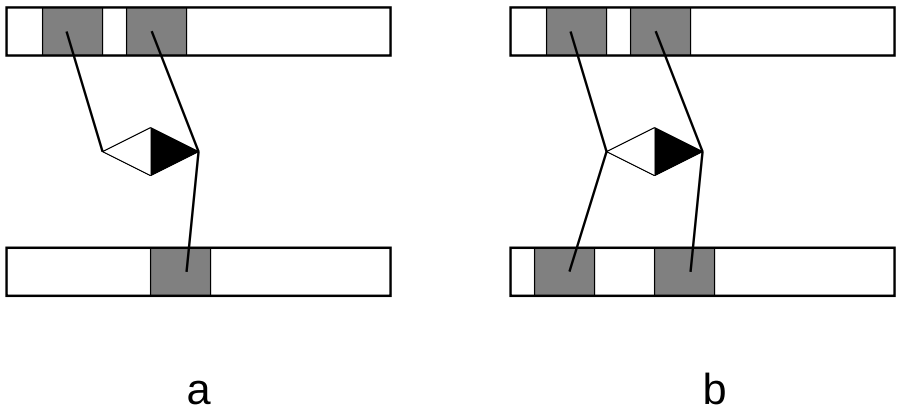

Figure 10. Two kinds of cross-dimensional relational correlations. The correlation represented by the MRU can constrain the value along one of the dimensions for only one of the objects, leaving the value of the other object along that dimension unspecified (a) or it can constrain the value along both dimensions for both objects (b).

tend to be activated. There would also be MRUs for the various sorts of cross-dimensional relational correlations. These would also tend to be activated. Thus a relation instance takes the form of a set of activated object feature units and a set of activated MRUs representing relational correlations among the features. These activated MRUs would permit pattern completion if the input patterns have missing information. We believe that many familiar relational categories such as ON are actually learned in terms of cross-dimensional relational correlations of this type. Thus for ON, the relative location of the upper and lower boundaries of two objects, which seems to define the relation for us, correlates with the shape, size, and movability of the objects.

As elsewhere with relational knowledge, the "binding problem" is an issue for this approach. Each MRU needs two "micro-roles," analogous to roles in other approaches, and it needs a mechanism to insure that the inputs to its two micro-roles come from separate objects. For this we make use of synchronization, as in a number of other recent connectionist approaches to the binding problem (Hummel & Biederman, 1992; Hummel & Holyoak, 1997; Shastri & Ajjanagadde, 1993). The inputs to each micro-role should be synchronized with one another and out of phase with the inputs on the other micro-role. We explain the details of how this is implemented in the next section.

In sum, other approaches assume that relations are built out of symbols (or symbol-like vectors) representing the constituent objects and the relation term or role names. The MRU approach, on the other hand, treats relations as composed of micro-relations, each of which relates features of objects, rather than whole objects, to one another. This approach opens up the possibility of an explicit account of where relations come from, how they are grounded in perceptual input.

Relational Tasks

We have seen what kind of information is required in the representation of relations and how this information is encoded in various approaches. But representations by themselves solve nothing. A relational processing system should be capable of performing the following basic relational tasks, all of them analogous to the corresponding tasks with objects.

Inference/Completion

1. Within-Relation: Object Identity
An incomplete relation instance is presented, and the system accesses its representation of the relation instance and fills in the missing information. For example, given the question, "what is above the table?," it fills in a book.

2. Within-Relation: Object Feature
A relation instance is presented, and, based on its general knowledge of the relation, the system infers a property of one of the related objects. For example, given ON(BOOK, X), it infers FLAT-TOP(X).

3. Between-Relation
Given a relation instance, the system retrieves other relation instances which follow from it. For example, given ON(BOOK, X), it infers MORE-MOVABLE(BOOK, X).

Categorization

Given an input situation, the system categorizes it as an instance of an abstract relational category by assigning it an explicit label. For example, given a scene with a particular book and a particular table, it outputs ON(BOOK,TABLE).

Imaging

Given an internal representation of a relation instance, the system views its contents. In particular, if it is a spatial relation instance, an internal image of the instance is created.

Learning

1. Storing New Relation Instances
An input situation is stored in long-term memory as a new relation instance.

2. Generalizing over Stored Relation Instances
Multiple relation instances become associated with one another as instances of a more abstract relational category in such a way that a new similar instance is processed or categorized appropriately. Note that this does not necessarily require that the relational category be explicitly encoded.

Most of the approaches to relational knowledge discussed above address more than one of these tasks, though none addresses all of them. In particular, none of them handles the relationship between perception and relations, a relationship which is behind both categorization and imaging. Two further approaches to relations are relevant because, unlike the others discussed, they *are* concerned with how relations, at least spatial relations, are derived from perceptual input. Hummel & Biederman's JIM model (1992) is designed to take images of line drawings of three-dimensional objects, break them into shape primitives, and relate the shape primitives to each other spatially. On the basis of the shape primitives and their relations, the model can be trained to classify objects, though it does not learn the shape primitives or relations themselves. Regier has developed a model of the learning of spatial terms (1996). It takes as input scenes (or sequences of scenes) in which two objects have already been segregated and assigned figure or ground status. The input is processed by a hard-wired visual component which extracts the sort of information that appears to be relevant for the representation of spatial relations across languages. The model learns to use this information in ways that depend on the target language.

In each of these cases, spatial relations are in some sense grounded in perception. But in neither case can be these be viewed as general approaches to the learning and representation of relational knowledge since they only *categorize* relations. They say nothing about the other tasks: storing relation instances in memory; performing inference or completion; or viewing a relation instance, a process which operates in the opposite direction from categorization.

In this paper we are concerned with the multi-purpose access and use of relational knowledge. We will argue that the MRU approach, embedded in a constraint-satisfaction connectionist network, accommodates all of the relational tasks.

In the next section, we discuss an implementation of the MRU approach.

An Implementation of Micro-Relations

Architecture

We propose a network in which two kinds of processing units interact, **micro-object units (MOUs)**, each responsible for an object feature or a cluster of object features, and **micro-relation units (MRUs)**, each responsible for a micro-relation or a cluster of micro-relations.

As we saw in the last section, the set of tasks making use of relational knowledge involves both accessing relational knowledge given different sorts of inputs and accessing different kinds of knowledge given relational inputs. If we assume that the same system is responsible for performing these different tasks, as would be the case, for example, in a model which made use of shared hardware for vision and imagery (Kosslyn, 1994), then a constraint-satisfaction network permitting multi-dimensional processing is more appropriate than a feed-forward network.

MOUs and MRUs are associated with one another in a generalization of a continuous Hopfield network. All connections are bi-directional and symmetric. Processing consists in clamping portions of the network according to an input pattern and allowing the rest of the network to settle into a stable state.

Learning consists in adjusting the weights on the connections joining units in response to a set of training patterns. For the purpose of this paper, we consider only the case of **unsupervised learning**, for which there is simply a set of training patterns to be auto-associated rather than a set of inputs and associated targets. The learning algorithm is an adaptation of **contrastive Hebbian learning (CHL)** (Movellan, 1990). CHL takes place in two phases. During the **positive** phase, the training patterns are clamped in the network, the network is allowed to settle, and the weight updates are accumulated. During the **negative** phase, the clamped units are unclamped, the network is again allowed to settle, and weight updates are *subtracted* from those accumulated during the positive phase. When the network succeeds in maintaining each training pattern after it is unclamped, the behavior of the network during the two phases is identical, and the positive and negative weight changes cancel each other out.

The behavior of a Hopfield network is governed by an **energy function**. As the network settles in response to an input pattern, it seeks a local energy minimum. The energy function is the sum of two components. One of these implements the basic Hebbian relationship: energy decreases with the product of each weight and the activations of the units joined by the connection. The other component tends to drive activations back to their resting values (Movellan, 1990). We can perform gradient descent on the energy function by differentiating by each of the weights. Given a particular sort of unit input function, this yields a learning rule for the two weight update phases. Learning is Hebbian (anti-Hebbian during the negative phase): each weight is adjusted in proportion to the product of the activations of the two units joined by the connection. As we shall see below, the basic CHL algorithm needs to be modified in two ways: to accommodate unit phase angles and coupling and to implement constraints in the behavior of MRUs. See the Appendix for details.

Processing Units

Micro-Object Units

As we saw in the section on relational knowledge, a system that uses relational knowledge must solve the binding problem; that is, it must provide a means by which features belonging to different objects can be distinguished from one another. In a number of recent models, object feature units have, in addition to activation, another value. Units which are "synchronized" on this other dimension represent features of a single object (Hummel & Holyoak, 1997; Hummel & Biederman, 1992; Shastri & Ajjanagadde, 1993; Sporns et al., 1989). In our network, each MOU has a **relative phase angle**[3] in addition to an activation. Units with the same relative phase angle are part of the same object, and units with different relative phase angles belong to different objects.

The connection between each pair of MOUs and between each MOU and MRU has not only a weight but also an associated **coupling function**, a function of the difference in phase angles of the two units. The coupling function must be symmetric about 0, and its derivative must be anti-symmetric about 0; see the Appendix for why these constraints must hold. Both the activation and the phase angle of an MOU are potentially modified each time a unit is updated, and both depend on the coupling function on the weights into the unit. The activation function for both MOUs and MRUs is the familiar interactive activation rule (McClelland & Rumelhart, 1981).
If $h_i^t > 0$,

$$\Delta a_i^t = h_i^t(a_i^{max} - (a_i^{t-1} - D_i a_i^{t-1})) \tag{1}$$

Else,

$$\Delta a_i^t = h_i^t((a_i^{t-1} - D_i a_i^{t-1}) - a_i^{min}),$$

[3]Units in the network can be viewed as oscillators, but since their periods are identical, there is no reason to actually implement the oscillation. Thus each is characterized by a *relative* phase angle, which remains unchanged except when the unit is influenced by other units to which it is connected. For convenience, however, we shall refer to this value simply as "phase angle."

where h_i^t is the input to unit i at time t; a_i^t is the activation of unit i at time t; and a_i^{max}, a_i^{min}, and D_i are the constant maximum activation, minimum activation, and decay rate associated with i.[4]

The input to an MOU from other MOUs is given by

$$h_i^t = \sum_{j=1}^{n} a_j^t \cdot w_{ij} \cdot \Phi(\varphi_i^t - \varphi_j^t), \tag{2}$$

where n is number of OUs in the network, w_{ij} is the weight connecting units i and j, Φ is the inter-unit coupling function,[5] and ϕ_i^t is the phase angle of unit i at time t.

The change in phase angle to an MOU i due to other MOUs is given by

$$\Delta\varphi_i^t = \sum_{j=1}^{n} a_j^t \cdot w_{ij} \cdot \Phi'(\varphi_i^t - \varphi_j^t), \tag{3}$$

A stable state of the network is a state in which neither activations nor phase angles are changing.

The coupling function also enters into the weight update rule. For a connection joining two MOUs, the rule is

$$\Delta w_{ij}^t = L \cdot a_i^t \cdot a_j^t \cdot \Phi(\varphi_i^t - \varphi_j^t), \tag{4}$$

where L is a constant learning rate.

The coupling function used in the network is

$$\Phi(x) = \cos^2 \frac{x}{2}. \tag{5}$$

For positive weights, the system consisting of the two units with this coupling function has an attractor at the state where the units are in phase and a repeller at the state where they are out of phase. The two units excite each other at all phase angle differences except π radians. For negative weights, there is an attractor at the out-of-phase state and a repeller at the in-phase state, and the units inhibit each other except when they are perfectly out of phase.

Micro-Relation Units

We have seen how the micro-relation approach differs from other approaches to relations in treating the primitives out of which relations are composed of as themselves relational. These micro-relations are associated both with features of objects and with each other. Because they are relational, they must include role-like elements which we will refer to as **micro-roles**. An implementation of micro-relations as micro-relation units (MRUs) should satisfy the following constraints:

[4]Since time will be irrelevant for the remaining discussion, we will omit the time superscript in what follows.

[5]In the current implementation, we use the same coupling for all pairs of units.

Micro-relations and object features

1. Since each MRU represents a binary micro-relation, it should become activated to the extent that it receives input from features of two different objects at its micro-roles. Thus the activation of an MRU depends not only on the total input it receives but also on the distribution of this input between its two micro-roles and on the identities associated with the input.

2. An activated MRU should activate sets of object features associated with distinct objects. Thus each micro-role should cause the object feature units (MOUs) which are positively connected to it not only to be activated but to take on the identity of the object bound to that micro-role.

3. Associations between object feature units (MOUs) and MRUs should have a strength reflecting the relational correlation between the features, and this strength should be learnable.

Micro-relations and other micro-relations

1. Associations between MRUs, representing relational correlations, should be constrained by the identities of the objects related by the MRUs. For example, an MRU representing the location of two objects could activate an MRU representing the meaning of the word *above* by causing the FIGURE micro-role of the *above* MRU to take on the identity of the object that is associated with the HIGHER micro-role of the location MRU. And two positively connected MRUs whose micro-roles are associated with completely different pairs of objects should fail to activate one another.

2. An association between MRUs should have a strength reflecting the correlation between the two micro-relations, and this strength should be learnable.

Like MOUs, MRUs have an activation reflecting the degree to which the micro-relation characterizes the current input. Also like MOUs, MRUs make use of relative phase angles to distinguish objects from one another. But, unlike MOUs, each MRU requires two relative phase angles, one for each micro-role. The total input to an MRU is the sum of the inputs to its two micro-roles.

Consider first an MRU which interacts only with MOUs. Each MOU may connect to either micro-role of a given MRU. The micro-role should couple with its MOU inputs just as MOUs couple with each other. That is, all else being equal, an MRU micro-role should have the mean of the phase angles of the MOUs it is positively connected to. Conversely, an activated MRU should tend to pass on the phase angles of its micro-roles to the MOUs with which they are connected. Thus with respect to input and coupling, each micro-role resembles a single MOU unit. These two properties of the MOU-MRU relationship are illustrated in Figure 11a and b.

At the same time, an MRU should fail to become activated to the extent that the inputs to its two micro-roles come from the same object. We achieve this property in part by associating the micro-roles of an MRU with their own MRU-internal coupling function, which tends to drive the micro-roles apart. This function is just the negative of the inter-unit coupling function we current use:

$$\Phi_r(x) = -\cos^2 \frac{x}{2}. \tag{6}$$

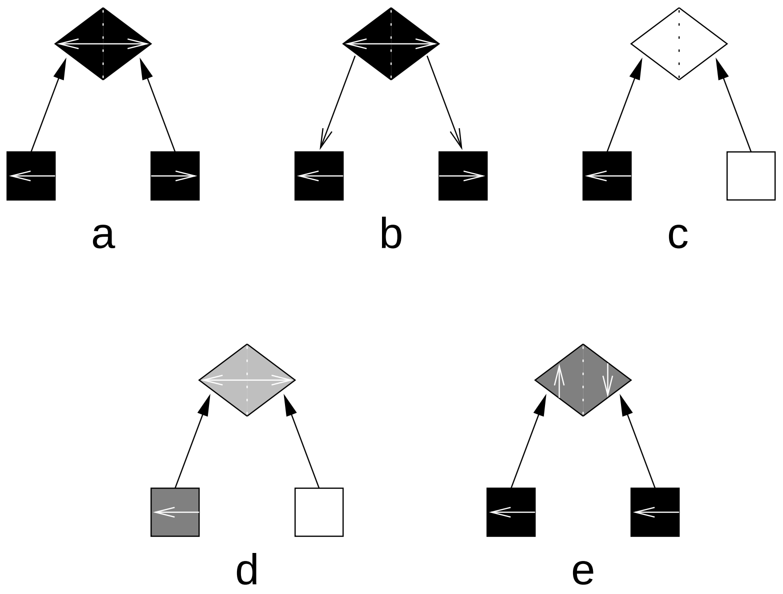

Figure 11. Relationships between micro-object and micro-relation units. MOUs are squares, MRUs are diamonds, phase angles are arrows, activation is indicated by darkness, and the direction of activation is indicated by the arrows connecting the units. (a) Two out-of-phase MOUs activate an MRU. (b) An MRU activates two MOUs. (c) Input from a single MOU fails to activate an MRU. (d) In-phase input from two MOUs weakly activates an MRU. (e) Weak input from a single MOU weakly activates an MRU.

For each MRU i there is a term in the energy function which includes this coupling function:

$$-a_i \cdot I \cdot \Phi_r(\phi_{i,L} - \phi_{i,R}), \tag{7}$$

where I is a constant controlling the relative strength of the internal coupling term and $\phi_{i,L}$ and $\phi_{i,R}$ are the left and right micro-role phase angles of unit i. This term is positive, that is, energy is increased, whenever the micro-roles of the MRU are not out-of-phase. The effect of these internal coupling terms is to cause the two micro-role phase angles of an MRU to repel one another and to inhibit MRUs to the extent that their phase angles are not out of phase.

There is still the further constraint that an MRU should be activated only if it receives input to both micro-roles. This situation is illustrated in Figure 11c. To achieve this, we add a further term to the energy function for each MRU which penalizes MRUs to the extent that the inputs to their two micro-roles are different. Since the addition of these terms (as well as the internal coupling terms) results in a more complex input function, we express the constraint in terms of the more basic “simple input.” The simple input to the left micro-role of MRU i is

$$h^*_{i,L} = \sum_j w_{i,L;j} \cdot a_j \cdot \Phi_r(\varphi_{i,L} - \varphi_j), \tag{8}$$

where $w_{i,L;j}$ is a weight connecting the left micro-role of MRU i to either an MOU or the micro-roles of an MRU, and φ_j is the phase angle of that MOU or MRU micro-role. The

micro-role asymmetry term for MRU i is

$$a_i A(h^*_{i,L} - h^*_{i,R})^2, \tag{9}$$

where A is a constant controlling the relative strength of this constraint. This term increases the energy to the extent that the simple input into the micro-roles of the MRU differs. The effect is to inhibit MRUs for which the simple micro-role inputs differ.

The addition of internal coupling and micro-role asymmetry terms to the energy equation results in additional terms in the input and phase angle update rules for MRUs and for MOUs which are connected to MRUs and also for the weight update rules for connections with an MRU on one or both ends. The complete rules are given in the Appendix.

These additional terms do not completely solve the problem of MRUs with input to only one micro-role or with in-phase input to the two micro-roles. First, the punishment accorded MRUs by the micro-role asymmetry terms depends on the the difference between the inputs to the two micro-roles rather than on their values. Thus a unit with no input to one micro-role and relatively small input to the other still receives some activation. This possibility is illustrated in Figure 11d. Second, the effect of the internal coupling terms together with relatively equal in-phase input to the two micro-roles can result in a weakly activated MRU whose micro-role phase angles are $\pi/2$ out of phase from the input phase angles. This possibility is illustrated in Figure 11e. Thus the current implementation can still be improved on, possibly through the use of a more complex micro-role asymmetry term.

Now consider how MRUs should interact with one another. MRUs must be connected to each other in such a way that they can pass on their phase angles, so they clearly require more than one connection joining them. In our network, each pair of connected MRUs is joined by two pairs of connections, one parallel (left-to-left and right-to-right micro-roles) and one opposing (left-to-right and right-to-left micro-roles). Each pair of connections is constrained to have the same weight. There are four possibilities for how two MRUs are connected and for what this signifies:

1. The two micro-relations have no effect on each other. In this case both weights are 0.

2. The two micro-relations correlate negatively with one another; that is, the presence of one leads one to expect the absence of the other. In this case both weights are negative.

3. The two micro-relations correlate positively with one another, and corresponding micro-roles are bound to the same object. In this case parallel micro-roles are connected by a positive weight, and the opposing weight is either zero or negative.

4. The two micro-relations correlate positively with one another, and opposing micro-roles are bound to the same object. In this case opposing micro-roles are connected by a positive weight, and the parallel weight is either zero or negative.

These four possibilities are illustrated in Figure 12.

The complete input, phase angle update, and weight update rules for MRU-MRU connections are given in the Appendix.

The framework described in this section exhibits most of the properties we have outlined for a system which learns and makes use of basic relational knowledge. It can represent the distinction between different objects at the level of features and at the level

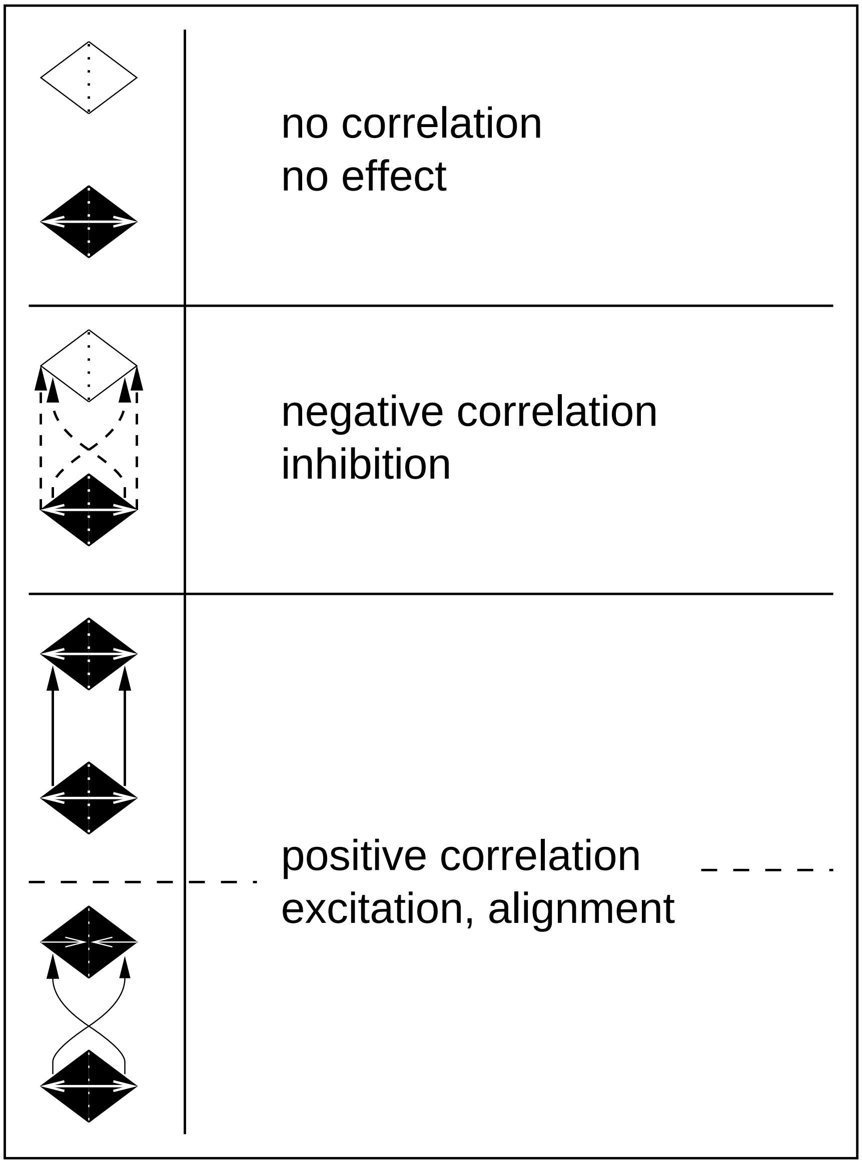

Figure 12. Possible relationships between micro-relation units. Phase angles of micro-roles are indicated by arrows. Direction of activation is indicated by the arrows representing the connections. Solid lines are positive connections; dashed lines are negative connections.

of micro-relations, "bind" object features to the micro-roles of micro-relations, and map micro-roles of different micro-relations onto one another.

In the next section we illustrate the properties of this approach in two simple simulations.

Simulations

In this section we describe several simulations designed to demonstrate that the MOU-MRU architecture fulfills some of the basic properties of a system which learns to use relational knowledge.

Simulation 1: Categorization and Imaging

Among the tasks that a system which handles relational knowledge should perform, relational categorization is probably the most fundamental. Within the MRU framework, relational categorization begins with input in the form of features of two objects and assigns a label to the relation between the objects. The reverse task, imaging, takes the label and possibly also a partial specification of the sensory/perceptual input and yields a completed sensory/perceptual pattern. One of the claims of the model is that the same connections should enable both of these behaviors.

To demonstrate these capacities, we trained a network to make the association between pairs of object features and relation categories. The network, shown in Figure 13, consisted of a layer of MOUs representing a single dimension, an intermediate layer of "hidden" MRUs, and a layer of category MRUs. There was complete connectivity between the connected layers. The network began with a set of weights which associated each pair of MOUs with a single hidden MRU. Other MOU-to-hidden weights were negative. The hidden-to-category and hidden-to-hidden weights began at 0.

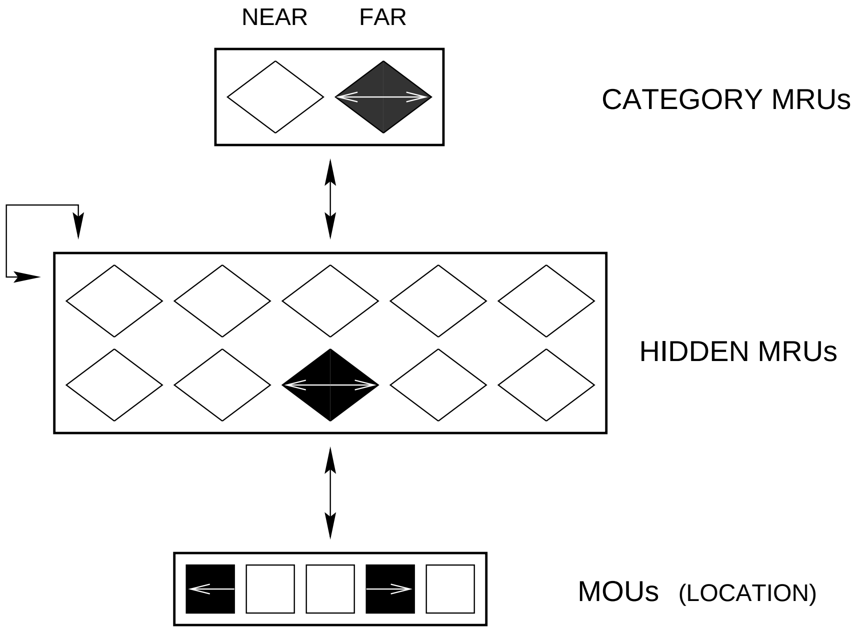

Figure 13. Architecture of the network used in Simulation 1. The network consists of three layers: a layer of MOUs representing the dimension LOCATION, a hidden layer of MRUs, and a layer of category MRUs representing the categories NEAR and FAR. As shown by the arrows, there is complete connectivity between the MOU and hidden layers, between the hidden and category layers, and within the hidden layer.

All possible pairs of activated MOUs were included in the training patterns. Each pattern was assigned to either the "near" or "far" category, depending on the distance between the activated units along the MOU dimension. Training was in unsupervised mode: during the positive learning phase, both the MOU and category layers were clamped, and during the negative learning phase, all units were unclamped. That is, the network was not learning to treat either layer as input or output but rather to auto-associate patterns across both layers. Following five repetitions of the training set, the network was tested in both the categorization and imaging directions. In the categorization direction, the MOU layer was clamped so that one pair of units was on with opposing phase angles. Performance was evaluated at the category layer. In the imaging direction, the category layer was clamped to one or the other category, one of the MOUs was clamped to agree with one of the category phase angles, and performance was evaluated over the unclamped MOUs. Thus, an imaging test corresponded to a question of the form, "if object Y is near an object at position X, where would Y be?" Table 2 shows the performance of the network, averaged over all of the test patterns. For the imaging task, there is sometimes more than one correct unit; in these cases, the most highly activated of these units was recorded.

Table 2: Simulation 1: Categorization and imaging. This table shows the average activation for correct and incorrect units and the average phase angle error (in radians) for correct units on two pattern completion tasks: categorization (MOU input, category output) and imaging (category and partial MOU input, MOU output).

	Correct unit		Incorrect units
	Activation	PA error	Activation
Categorization	0.703	.299	0.00
Imaging	0.847	0.122	0.023

Simulation 2: MR Correlations

A key feature of the model we are proposing is that abstract relations such as ON are actually composed of correlations among micro-relations between features of the related objects. Thus the MOU-MRU architecture should at a minimum have the capacity to learn correlations between MRUs.

To illustrate this property of the model, we trained a network to associate MRUs on one dimension with those on another. The network consisted of two layers of MOUs, one for each dimension, and two layers of MRUs, one associated with each object dimension. The two MRU layers were connected to each other, but the two MOU layers were not. Again the network began with a set of weights which associated each pair of MOUs in an input layer with a single MRU. Figure 14 shows the network architecture.

The network was trained on patterns which associated pairs of MOUs on one dimension with identical pairs on the other dimension. Note that since it is pairs of out-of-phase MOUs which are associated with one another, these are relational correlations, requiring MRUs to be learned.[6] Training was again unsupervised: neither MOU layer was treated as

[6] In any case, there are no direct connections between the two MOU layers, so the correlations can only

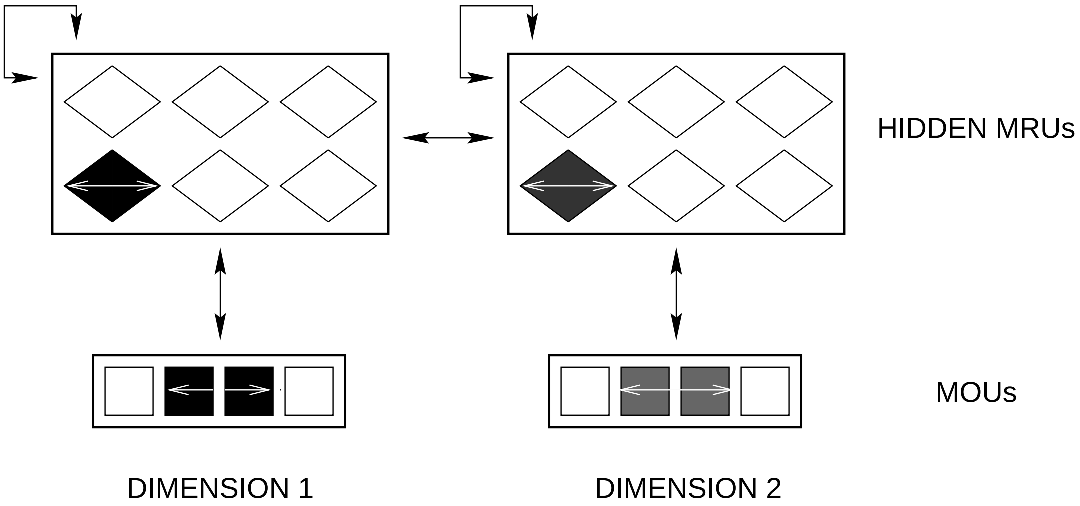

Figure 14. Architecture of the network used in Simulation 2. The network consists of four layers: two layers of MOUs representing the two correlated dimensions and two hidden layers of MRUs, each associated with one dimension. As shown by the arrows, there is complete connectivity between the two hidden MRU layers, between each MRU layer and its corresponding MOU layer, and within each MRU layer.

input or output; the network was simply learning to auto-associate the patterns. Table 3 shows the performance of the network, following 15 repetitions of the training patterns, on pattern completion tasks involving the presentation of a pattern on one of the MOU layers.

Table 3: Simulation 2: MRU correlations. This table shows the average activation for correct and incorrect units and the average phase angle error (in radians) for correct units on a pattern completion task.

Correct units		Incorrect units
Activation	PA error	Activation
0.807	0.729	0.00

Implications and Predictions

The MRU approach to representation, learning, and use of relational knowledge differs from alternative approaches in several important ways. First, there are no explicit roles or relation terms, although linguistic labels (words) could be associated with clusters of MRUs to provide a pointer to a relation. Since roles and relation terms are not assumed, it is possible to study how they develop out of object and relation micro-features. This also allows us to study how different languages, providing different labels, affect the relational clusters that are based on perception only.

be learned through the connections between the two MRU layers.

Second, relations are built out of micro-relations, each of which relates object features rather than whole objects. Thus coherent objects are not necessary for relations to begin to be activated. This feature of the model permits the investigation of the interaction between ways in which objects and relations are learned and activated. For example, since relations do not need to wait for their object arguments to be fully activated, an activated relational category could influence the way in which objects are categorized. Thus, the model predicts that in an ambiguous situation construed as IN, the lower object will be categorized as a CONTAINER, while this will not be the case if the situation is construed as ON.

Third, in our framework there is no distinction between relations between individual objects and relations between categories of objects. The level of abstraction at which information is represented depends on the number of MRUs that are involved in representing it, and this varies as more instances of a relation are seen. Two predictions follow from this fact and from the nature of the learning mechanism. The first concerns the developmental course of relations. Each relation should start in a relatively context-specific form, tied to particular pairs of objects or particular kinds of objects. Later, as MRUs become associated with one another, the relation becomes more abstract. If this is true, we expect children to have more context-specific relations than adults. The second prediction concerns the developmental course of relational similarity. Since more complex relations are built out of correlations between simpler relations, the tendency to favor relational over superficial similarity will correlate with the increasing abstractness of relations. This relational similarity advantage could be independent for each relation, depending on how much experience the child has had with it.

Finally, in our framework all relational knowledge is ultimately correlational. Therefore, a major cause of the difference in ease of learning of two relations should be the difference in the correlational structure behind the relations. For example, this helps explain why LEFT and RIGHT are more difficult than ON and UNDER. While there is much more to ON than just vertical orientation, there is little more than horizontal orientation to LEFT. Concepts such as LEFT can be made easier through training that clearly establishes additional correlations, for example, with handedness (Clark, 1973).

Conclusions

As is shown by our use of language and the kinds of inferences we can make, we have the ability to represent and use relational knowledge. To understand this ability, we need to take into consideration two often neglected facts about relational knowledge. First, relational knowledge is learned. We must understand how it is acquired from experience, how the representation of relational knowledge changes over time and with more experience, and how it is grounded in perception and action. Second, people use relational knowledge to perform a variety of tasks: inference, classification, prediction, imaging. A system that is especially wired to perform one task or another will be missing important aspects of what we want to model.

We have presented a framework that addresses these issues by considering relations to be correlations of micro-features and micro-relations. We have shown that this system can acquire, represent, and use representational correlations, that is, that it performs the kinds of tasks people perform using relational knowledge.

The simple tasks we have presented in this paper are far from the sophisticated behaviors exhibited by other models of relational knowledge. However, we believe that real insight into complex tasks such as analogy and metaphor must rest on an understanding of more fundamental issues: what relations really are, how they arise, and how they are used to perform the sorts of relational tasks that a four-year-old faces. We believe that our model is a first step in this direction.

Appendix: Mathematical Details of the Model

Micro-Object Units

In this section we show how Contrastive Hebbian Learning (CHL) (Movellan, 1990) needs to be modified to accommodate units with relative phase angles. We follow the derivation in Movellan closely.

Movellan defines a continuous Hopfield energy function

$$F = E + S \tag{10}$$

where E reflects the constraints imposed by the weights in the network and S the tendency to drive the activations to a resting value. For our network S is the same as for a network with no phase angles:

$$S = \sum_{i=1}^{n+m} \int_{rest_i}^{a_i} f_i^{-1}(a)da \tag{11}$$

where $n+m$ is the number of units in the network, a_i is the activation of unit i, f_i is the activation function for unit i, and $rest_i = f(0)$.
However, E for MOUs becomes

$$E^{MOU} = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i \cdot w_{ij} \cdot a_j \cdot \Phi(\varphi_i - \varphi_j) \tag{12}$$

where w_{ij} is the weight connecting units i and j and Φ is the inter-unit coupling function. In what follows we will abbreviate $\Phi(\varphi_i - \varphi_j)$ as $\Phi_{j;i}$.

The coupling function must be differentiable and satisfy the following:

$$\Phi_{i;j} = \Phi_{j;i} \tag{13}$$

$$\Phi'_{i;j} = -\Phi'_{j;i} \tag{14}$$

When the network is stable, the inverse of the activation function for each unit is equal to the input into that unit:

$$f_i^{-1}(\breve{a}_i) = \breve{h}_i = \sum_{j=1}^{n} \breve{a}_j w_{ij} \breve{\Phi}_{j;i} \tag{15}$$

where ($\breve{\ }$) represents equilibrium and h_i is the input to unit i. Furthermore, when the network is stable, the phase angle of each unit no longer changes:

$$\Delta\breve{\varphi}_i^{MOU} = \sum_{j=1}^{n} \breve{a}_j w_{ij} \breve{\Phi}'_{j;i} = 0 \tag{16}$$

Movellan defines the contrastive function J as

$$J = \breve{F}^{(+)} - \breve{F}^{(-)} \tag{17}$$

and shows that the CHL rule minimizes J. We follow his derivation for the case where units have phase angles.

The energy of the network for MOUs at equilibrium is

$$\breve{E}^{MOU} = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \breve{a}_i w_{ij} \breve{a}_j \breve{\Phi}_{j;i} \tag{18}$$

Extracting the terms with a w_{ij} term,

$$\breve{E}^{MOU} = -\frac{1}{2}\left(2\breve{a}_i w_{ij} \breve{a}_j \breve{\Phi}_{j;i} + \sum_{k=1}^{n} \sum_{\substack{l=1 \\ k,l\neq i,j;k,l\neq j,i}}^{n} \breve{a}_k w_{kl} \breve{a}_l \breve{\Phi}_{l;k}\right) \tag{19}$$

Differentiating with respect to a single weight w_{ij} and considering that w_{ij} is the only weight depending on w_{ij},

$$\begin{aligned} \frac{\partial \breve{E}^{MOU}}{\partial w_{ij}} &= -\frac{1}{2}\Bigg[2\breve{a}_i\breve{a}_j\breve{\Phi}_{j;i} + 2w_{ij}\breve{a}_i\breve{\Phi}_{j;i}\frac{\partial \breve{a}_j}{\partial w_{ij}} + 2w_{ij}\breve{a}_j\breve{\Phi}_{j;i}\frac{\partial \breve{a}_i}{\partial w_{ij}} + \\ & \breve{a}_i w_{ij}\breve{a}_j\left(\breve{\Phi}'_{j;i}\left(\frac{\partial\breve{\varphi}_i}{\partial w_{ij}} - \frac{\partial\breve{\varphi}_j}{\partial w_{ij}}\right) + \breve{\Phi}'_{i;j}\left(\frac{\partial\breve{\varphi}_j}{\partial w_{ij}} - \frac{\partial\breve{\varphi}_i}{\partial w_{ij}}\right)\right) + \\ & \sum_{k=1}^{n}\sum_{\substack{l=1 \\ k,l\neq i,j;k,l\neq j,i}}^{n} w_{kl}\left(\breve{a}_k\breve{\Phi}_{l;k}\frac{\partial\breve{a}_l}{\partial w_{ij}} + \breve{a}_l\breve{\Phi}_{l;k}\frac{\partial\breve{a}_k}{\partial w_{ij}} + \breve{a}_k\breve{a}_l\breve{\Phi}'_{l;k}\left(\frac{\partial\breve{\varphi}_k}{\partial w_{ij}} - \frac{\partial\breve{\varphi}_l}{\partial w_{ij}}\right)\right)\Bigg] \end{aligned} \tag{20}$$

From Equation 14, we have

$$\begin{aligned} &\breve{a}_i w_{ij}\breve{a}_j\left(\breve{\Phi}'_{j;i}\left(\tfrac{\partial\breve{\varphi}_i}{\partial w_{ij}} - \tfrac{\partial\breve{\varphi}_j}{\partial w_{ij}}\right) + \breve{\Phi}'_{i;j}\left(\tfrac{\partial\breve{\varphi}_j}{\partial w_{ij}} - \tfrac{\partial\breve{\varphi}_i}{\partial w_{ij}}\right)\right) \\ &= 2w_{ij}\breve{a}_i\breve{a}_j\breve{\Phi}'_{j;i}\tfrac{\partial\breve{\varphi}_i}{\partial w_{ij}} + 2w_{ij}\breve{a}_i\breve{a}_j\breve{\Phi}'_{i;j}\tfrac{\partial\breve{\varphi}_j}{\partial w_{ij}} \end{aligned} \tag{21}$$

and

$$\sum_{k=1}^{n}\sum_{\substack{l=1 \\ k,l\neq i,j;k,l\neq j,i}}^{n} w_{kl}\left(\breve{\Phi}'_{l;k}\left(\frac{\partial\breve{\varphi}_k}{\partial w_{ij}} - \frac{\partial\breve{\varphi}_l}{\partial w_{ij}}\right)\right) = 2\sum_{k=1}^{n}\sum_{\substack{l=1 \\ k,l\neq i,j;k,l\neq j,i}}^{n} w_{kl}\breve{a}_k\breve{a}_l\breve{\Phi}'_{l;k}\frac{\partial\breve{\varphi}_k}{\partial w_{ij}} \tag{22}$$

Substituting these into Equation 20,

$$\frac{\partial \breve{E}^{MOU}}{\partial w_{ij}} = -\frac{1}{2}\left(2\breve{a}_i\breve{a}_j\breve{\Phi}_{j;i} + 2\sum_{k=1}^{n}\frac{\partial \breve{a}_k}{\partial w_{ij}}\sum_{l=1}^{n} w_{kl}\breve{a}_l\breve{\Phi}_{l;k} + 2\sum_{k=1}^{n}\breve{a}_k\frac{\partial \breve{\varphi}_k}{\partial w_{ij}}\sum_{l=1}^{n} w_{kl}\breve{a}_l\breve{\Phi}'_{l;k}\right) \tag{23}$$

From 15 and 16, we have the following for the case where $i \neq j$. Since there are no self-recurrent connections in our network, we need only consider this case.

$$\frac{\partial \breve{E}^{MOU}}{\partial w_{ij}} = -\breve{a}_i\breve{a}_j\breve{\Phi}_{j;i} - \sum_{k=1}^{n}\breve{h}_k\left(\frac{\partial \breve{a}_k}{\partial w_{ij}}\right) - \sum_{k=1}^{n}\Delta\breve{\varphi}_k\breve{a}_k\frac{\partial \breve{\varphi}_k}{\partial w_{ij}} \tag{24}$$

From 16, the last term is 0, and we have

$$\frac{\partial \breve{E}}{\partial w_{ij}} = -\breve{a}_i\breve{a}_j\breve{\Phi}_{j;i} - \sum_{k=1}^{n}\breve{h}_k\left(\frac{\partial \breve{a}_k}{\partial w_{ij}}\right) \tag{25}$$

From Equation 11,

$$\frac{\partial \breve{S}}{\partial w_{ij}} = \sum_{k=1}^{n} f_k^{-1}(\breve{a}_k)\frac{\partial \breve{a}}{\partial w_{ij}} \tag{26}$$

and from Equation 15, we have

$$\frac{\partial \breve{F}}{\partial w_{ij}} = -\breve{a}_i\breve{a}_j\breve{\Phi}_{j;i} \tag{27}$$

making

$$\frac{\partial \breve{J}}{\partial w_{ij}} \propto \breve{a}_i^{(+)}\breve{a}_j^{(+)}\breve{\Phi}_{j;i}^{(+)} - \breve{a}_i^{(-)}\breve{a}_j^{(-)}\breve{\Phi}_{j;i}^{(-)} \tag{28}$$

which shows that the modified CHL rule

$$\Delta w_{ij}^{MOU} \propto \breve{a}_i^{(+)}\breve{a}_j^{(+)}\breve{\Phi}_{j;i}^{(+)} - \breve{a}_i^{(-)}\breve{a}_j^{(-)}\breve{\Phi}_{j;i}^{(-)} \tag{29}$$

descends in the J function.

Micro-Relation Units

Unlike MOUs, MRUs have two phase angles, one for each micro-role. There are two constraints that affect these phase angles.

1. Each MRU has an internal coupling function which tends to push its two phase angles apart.

2. Each MRU is punished to the extent that the simple input into its two micro-roles is different.

Each of these constraints adds a set of terms to the energy equation, and when the equation is differentiated with respect to the weights, modified input and phase angle update rules are required in order for the derivatives to drop out of the right side of the equation for MRUs that corresponds to Equation 23. The resulting weight update rules also differ.

Here we omit the derivation since it is similar to that given in the previous section; we give only the resulting rules.

For MOUs connected to MRUs, the input and phase angle update rules due to the MRUs are:

$$\begin{aligned} h_i^{MOU \leftarrow MRU} &= \sum_{j=n+1}^{n+m} a_j(1 - 2A(h^*_{j,L} - h^*_{j,R})) \\ &\quad [w_{i;j,L}\Phi_{j,L;i} - w_{i;j,R}\Phi_{j,R;i}] \end{aligned} \tag{30}$$

$$\begin{aligned} \Delta\varphi_i^{MOU \leftarrow MRU} &= \sum_{j=n+1}^{n+m} a_j(1 - 2A(h^*_{j,L} - h^*_{j,R})) \\ &\quad \left[w_{i;j,L}\Phi'_{j,L;i} - w_{i;j,R}\Phi'_{j,R;i}\right], \end{aligned} \tag{31}$$

where n is the number of MOUs in the network, m is the number of MRUs in the network, A is the constant controlling the relative strength of the micro-role asymmetry constraint, L and R subscripts index the left and right micro-roles of MRUs, and $h^*_{j,L}$ and $h^*_{j,R}$ are the simple inputs to the left and right micro-roles of the MRU j, as given above in Equation 8.

For MRUs, the input and phase angle update rules are:

$$\begin{aligned} h_i^{MRU} &= \sum_{j=1}^{n} a_j(w_{i,L;j}\Phi_{j;i,L} + w_{i,R;j}\Phi_{j;i,R}) + \\ &\quad \sum_{j=n+1}^{n+m} a_j(1 - 2A(h^*_{j,L} - h^*_{j,R})) \\ &\quad (w_{P|i;j}(\Phi_{j,L;i,L} - \Phi_{j,R;i,R}) + w_{O|i;j}(\Phi_{j,R;i,L} - \Phi_{j,L;i,R})) - \\ &\quad A(h^*_{i,L} - h^*_{i,R})^2 + I\Phi_{r;i} \end{aligned} \tag{32}$$

$$\begin{aligned} \Delta\varphi_{i,L}^{MRU} &= \sum_{j=1}^{n} a_j w_{i,L;j}\Phi'_{j;i,L} + \\ &\quad \sum_{j=n+1}^{n+m} a_j(.5 + 2A(h^*_{j,L} - h^*_{j,R}))(w_{P|i;j}\Phi'_{P|j,L;i,L} - w_{O|i;j}\Phi'_{O|j,R;i,L}) + \\ &\quad I\Phi'_{r;i}, \end{aligned} \tag{33}$$

where $w_{P|i;j}$ and $w_{O|i;j}$ are the weights on the parallel and opposing connections joining MRUs i and j, I is the constant controlling the relative strength of the internal coupling constraint and $\Phi_{r;i}$ is the internal coupling function applied to the difference in phase angles of MRU i.

For connections joining MOUs and MRUs, the weight update rule for the left micro-role is:

$$\Delta w_{i;j,L} = La_i a_j \Phi_{j,L;i}(1 - 2A(h^*_{j,L} - h^*_{j,R})), \tag{34}$$

where L is the learning rate.

For connections joining pairs of MRUs, the weight update rule for the parallel connection is:

$$\Delta w_{P|i;j} = La_i a_j(.5 - 2A)[(h^*_{j,L} - h^*_{j,R}) + (h^*_{i,L} - h^*_{i,R})](\Phi_{j,L;i,L} + \Phi_{j,R;i,R}). \qquad (35)$$

References

Clark, H. H. (1973). Space, time, semantics, and the child. In T. E. Moore (Ed.), *Cognitive development and the acquisition of language* (pp. 27–63). New York: Academic Press.

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

Halford, G. S., Wilson, W. H., Guo, J., Gayler, R. W., Wiles, J., & Stewart, J. E. M. (1994). Connectionist implications for processing capacity limitations in analogies. In K. J. Holyoak & J. Barnden (Eds.), *Advances in connnectionist and neural computation theory, vol. 2: Analogical connections* (pp. 363–415). Norwood, NJ: Ablex.

Halford, G. S., Wilson, W. H., & Phillips, S. (1998). Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology. *Behavioral and Brain Sciences*, *21*, 00–00.

Hummel, J. E., & Biederman, I. (1992). Dynamic binding in a neural network for shape recognition. *Psychological Review*, *99*, 480–517.

Hummel, J. E., & Holyoak, K. J. (1997). Distributed rperesentation of structure: A theory of analogical access and mapping. *Psychological Review*, *104*, 427–466.

Kersten, A. W., & Billman, D. (1997). Event category learning. *Journal of Experimental Psychology: Learning, Memory and Cognition*, *23*(3), 658–???

Kosslyn, S. M. (1994). *Image and brain: The resolution of the imagery debate.* Cambridge, MA: MIT Press.

Langacker, R. (1987). Nouns and verbs. *Language*, *63*, 53–94.

McClelland, J., & Rumelhart, D. (1981). An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, *88*, 375–107.

Movellan, J. (1990). Contrastive Hebbian learning in the continuous Hopfield model. In D. Touretzky, J. Elman, T. Sejnowski, & G. Hinton (Eds.), *Proceedings of the 1990 Connectionist Models Summer School* (pp. 10–17). San Mateo, CA: Morgan Kaufmann.

Pearce, J. M. (1994). Discrimination and categorization. In N. J. Mackintosh (Ed.), *Animal learning and cognition, handbook of perception and cognition* (2nd ed.). Academic Press.

Phillips, S., Halford, G. S., & Wilson, W. H. (1995). The processing of associations versus the processing of relations: a systematic comparison. In J. D. Moore & J. R. Lehman (Eds.), *Proceedings of the annual conference of the cognitive science society* (Vol. 17, pp. 688–691).

Plate, T. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, *6*, 623–641.

Regier, T. (1996). *The human semantic potential: Spatial language and constrained connectionism.* Cambridge, MA: MIT Press.

Rosch, E. H., Mervis, C. B., Gray, W. D., Johnson, D. M., & Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology*, *8*, 382–439.

Shastri, L., & Ajjanagadde, V. (1993). From simple associations so systematic reasoning: A connectionist representation of rules, variables, and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences*, *16*, 417–494.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, *46*, 159–216.

Sporns, O., Gally, J. A., George N. Reeke, J., & Edelman, G. M. (1989). Reentrant signaling among simulated neuronal groups leads to coherency in their oscillatory activity. *Proceedings of the National Academy of Sciences*, *86*, 7265–7269.

Tesar, B. B., & Smolensky, P. (1994). Synchronous firing variable binding is a tensor product representation with temporal role vectors. In *Annual conference of the cognitive science society* (Vol. 16, pp. 870–875).