A Symbolic-Connectionist Model of Relation Discovery

Leonidas A. A. Doumas (adoumas@psych.ucla.edu)
John E. Hummel (jhummel@psych.ucla.edu)
Department of Psychology, University of California, Los Angeles
405 Hilgard Ave.
Los Angeles, CA 90095-1563

Abstract

Relational reasoning is central in human cognition. Numerous computational models address the component processes of relational reasoning, however these models require the modeler to hand-code the vocabulary of relations on which the model operates. The acquisition of relational concepts remains poorly understood. We present a theory of relation discovery instantiated in a symbolic-connectionist model, which learns structured representations of attributes and relations from unstructured distributed representations of objects by a process of comparison, and subsequently refines these representations through a process of mapping-based schema induction.

Keywords: relations, learning, neural network, symbolic processing, structured representations

Relational Reasoning

Virtually every conscious thought you have expresses a relation. From the mundane, like "I'm late for work", to the sublime, like Cantor's proof that the cardinal number of the real numbers is greater than that of the integers, we are constantly representing and reasoning with relations. Relational thinking is so commonplace it is easy to take for granted, but the ability to form and manipulate relational representations appears late in human development (Gentner & Rattermann, 1991; Smith, 1989), and is a late evolutionary development that appears to distinguish human cognition from that of other animals (Holyoak & Thagard, 1995; Thompson & Oden, 2000).

An important theme that has emerged from the study of relational thinking – both empirical and theoretical – is that the kinds of problems a person (or model) can solve depend critically on what the person (or model) can and does represent. However, little empirical work, and almost no theoretical work, has addressed the problem of how we acquire relational concepts. Models based on relational representations (e.g., Falkenhainer, Forbus, & Gentner, 1989; Hummel & Holyoak, 1997, 2003) have made important strides elucidating the nature of relational thought. However, these models are all granted a vocabulary of relational representations by the modeler; they do not learn the relations they need for themselves. Although they address our capacity to manipulate relational representations, they do not address the question of where these representations come from in the first place.

Relations are Hard to Learn

Learning relational concepts is difficult for two reasons. The first begins with the very definition of a relation: A relation is a property that holds over a *collection* of arguments; it is never observable in a single object, so learning relations is vastly underconstrained by the examples from which we learn them. Take, for example, the relation *same-shape* (*x*, *y*). When universally quantified it takes any shape as input, and therefore its truth-value (i.e., whether *x* and *y* are the same shape) is completely uncorrelated with the specific visual features of any shapes (or any pair of shapes for that matter). As a result, it cannot be learned from the simple covariation of visual features.

The second difficulty stems from the properties of relational representations. Relational representations are structure sensitive and semantically rich (Hummel & Holyoak, 1997). In a relational expression the meaning of the individual relational roles and their fillers is invariant with their arrangement in the expression (i.e., they are independent), but the meaning of the expression as a whole is a function of both the elements that compose the expression and their arrangement (i.e., the bindings of fillers to relational roles). Consider the statements *chase* (Bill, Joe) and *chase* (Joe, Bill). We can appreciate that they mean different things (even though they are composed of the same elements) because we can appreciate that the bindings of objects to relational roles is reversed in the two statements. We can also appreciate that the individual elements *chase*, Joe, and Bill, mean the same things in both statements (despite the fact that they are in different compositions). Additionally, we can cast these elements in novel configurations, for example generalizing the *chase* relation to novel arguments (e.g. *chase* (spoon, sprocket)). Thus, a relational concept must be represented independently of the examples from which it is learned, must be able to take arguments from both within and outside the set on which it was learned (i.e., we must be able to extrapolate the relation to novel values), and must specify the bindings of its arguments to its relational roles explicitly.

Relational representations also explicitly specify the semantic content of objects and relational roles (e.g., the *lover* and *beloved* roles of *love* (*x*, *y*) or the *liker* and *liked* roles of *like* (*x*, *y*)): We know what it means to be a lover, and that knowledge is part of our representation of the relation itself. Consequently, it is easy to appreciate that the patient (i.e., *killed*) role of *murder* (*x*, *y*) is like the patient *manslaughter* (*x*, *y*), even though the agent roles differ (i.e., the act is intentional in the former case but not the latter); and the agent role of *murder* (x, y) is similar to the agent role of *attempted-murder* (x, y), even though the patient roles differ.

A solution to the problem of learning representations of relations that have all these properties has proven elusive. Gasser and Colunga (2001) have made important strides toward modeling the learning of relational concepts. However the representations this model forms do not maintain independence of relations and arguments while explicitly specify the bindings of specific roles to specific fillers. Bindings are carried by connections between units, and connections only implicitly represent bindings (von der Malsburg, 1981/1994). Moreover, the model learns relations by learning correlations of specific feature values with relational labels. As noted above, one of the difficulties of learning truly relational representations is that they can be extrapolated to novel arguments and are, therefore, not learnable as covariations among features.

Constraints on Discovering Relations

Given these considerations, what kinds of cognitive operations can lead to the discovery of new relational concepts? One step toward constraining this otherwise deeply underconstrained problem is to choose an appropriate form of knowledge representation. One such representation is a role-filler binding scheme, in which relational roles and their arguments are represented explicitly and bound together to form role-filler sets (much like collections of single-place predicates). Collections of role-filler sets are linked together to form whole relational structures. For example, the representation of *loves* (John, Mary) consists of the representation of the *lover* role bound to John (*lover*(John)) and the *beloved* role bound to Mary (*beloved*(Mary)) linked together to form the structure (*lover*(John)+*beloved*(Mary)) (see e.g., Doumas & Hummel, in press).

Role-filler binding systems provide a natural constraint on the problem of relation discovery by reducing relational roles to single place predicates. As a result, the problem of learning relations reduces, at least in principle, to the problem of learning single-place predicates (i.e., object properties) and then linking those predicates to form complete relational structures.

Given this, how might we learn single-place predicates (i.e., object properties)? An important theme that has emerged in the literature on relational reasoning is that comparison plays a central role in all forms of relational reasoning (see Gentner, 1983, 2003; Holyoak & Thagard, 1995). A primary hypothesis motivating the present research is that comparison may also play a central role in the discovery and predication of new relations. The reason is that comparison, by putting objects into direct contrast, might serve to highlight shared properties. By revealing shared attributes of otherwise different-seeming systems, comparison may bootstrap the discovery and explicit representation of object properties and subsequently link sets of shared properties together to form whole relational structures.

For example, consider what happens when a child learns the property "red" by comparing an apple to a toy fire engine. Assume that when the apple and fire engine are compared, any features they share will be highlighted (i.e., get more input than unshared features). As apples and fire-engines are both red, perhaps the feature "red" gets most active. If the child can predicate the highlighted feature (i.e., attach a predicate to the feature "red"), she will have formed an explicit representation the property "red". If she can then bind that property to the apple (or fire engine) she will have explicitly predicated the property of "red" about the apple (or fire engine). Consistent with this idea, several previous studies have demonstrated that structure mapping bootstraps the induction of abstract relational schemas (e.g., Gick & Holyoak, 1983; Ratterman & Gentner, 1998), and that comparison helps people appreciate what lower-order relations might be relevant to a specific task (Kotovsky & Gentner, 1996; Namy & Gentner, 2002; Sandhofer & Smith, 2001; Yamauchi & Markman, 1998).

In order for a system to perform comparison-based predication, its representation of roles and fillers must share a common basis (i.e., role and filler representations should share the same pool of features). To illustrate, consider our example from above. Initially "red" is an implicit feature of both the apple and the fire engine. When they are compared, the feature "red" is abstracted to a new representation (i.e., a new unit learns a connection to this feature), but the same feature codes "red" in both cases. If the representation of a property in an explicit form (i.e., as a predicate that takes an object as an argument) was coded by a different set of features than the representation of that property in a holistic form (i.e., as a feature of an object), then the system could not learn explicit representations by abstracting features out of holistic representations: The representation of a property as an implicit feature of an object would have nothing in common with the explicit representation of that property.

We now present a theory of relation discovery based on these constraining assumptions embodied in a computer simulation called *DORA* (Discovery Of Relations by Analogy).

The DORA Model

Representational Structure

Representations in DORA are an extension of those in Hummel and Holyoak's (1997, 2003) LISA. Like LISA, DORA represents propositions in a hierarchy of distributed and localist codes (see Figure 1). At the bottom of the hierarchy, *semantic units* code for the features of relational roles and their fillers in a distributed fashion. At the next level are localist *token* units that code for specific relational roles and objects. For example, the proposition "Bill chases Larry", would be represented (in part) by units for the relational roles *chaser* and *chased* and the objects Bill and Larry (see Figure 1). Bill would be connected to a set of semantic units denoting his features (e.g., "adult", "male", "police-officer") and Larry to a set of semantic units denoting his features (e.g., "adult", "male", "criminal"). Similarly, the *chaser* and *chased* roles would be connected to the semantic units denoting their features. Semantically related objects and relational roles (e.g., *chaser* and *pursuer*) share many semantic units, making their semantic similarity explicit. Above the role and object units *role-binding* (RB) units encode the bindings of specific roles to specific fillers. Continuing our example, one RB unit would code for the binding of Bill to the *chaser* role and another would code for the binding of Larry to the *chased* role. At the top of the hierarchy proposition (P) units bind sets of role-filler bindings into complete relational structures.

Role-Filler Binding

The hierarchy in Figure 1 represents a proposition in DORA's long-term-memory (LTM). When a proposition becomes active (i.e., enters working memory; WM) DORA uses a form of asynchronous binding (see Love, 1999) to bind roles to their fillers: Bound role-filler pairs fire in direct sequence, which serves to dynamically bind roles to fillers in WM, and also keeps the representations of roles and fillers distinct for the purposed of processing. To illustrate, in order to bind Bill to the *chaser* role and Larry to the *chased* role (and so represent *chase* (Bill, Larry)), the units corresponding to the *chaser* role fire directly followed by the units corresponding to Bill (see Figure 2a and b), then the units for the *chased* role fire directly followed by the units for Larry (Figure 2c and d). A system that is sensitive to couplets (or pairs) of activation can use this information to represent the bindings of Bill to the *chaser* role and Larry to the *chased* role.

Comparison Based Predication

Because DORA uses a common pool of semantic units to code the features of both roles and objects, it can learn predicates by comparing objects. Propositions in DORA are divided into two sets, a driver and a recipient set (see Hummel & Holyoak, 1997). Token units in the driver become active and pass activation to their semantics. Through the shared semantic pool, units in the recipient propositions become active and respond to these patterns of activation. Using a mapping algorithm adapted from LISA, token units of the same kind (e.g., role/filler, RB) that are active at the same time in both the driver and recipient develop excitatory *mapping connections* to one-another. These connections represent existing mappings and constrain the discovery of future mappings.

In DORA comparison is accomplished via this mapping process. When propositions are compared corresponding units develop excitatory mapping connections. Consider the example of our child learning the property "red". Because the apple and the fire engine were compared they are mapped. Thus, an excitatory connection develops between them and one tends to activate the other (Figure 3a). Both units also activate their semantic features. Shared semantics receive twice as much input and therefore become twice as

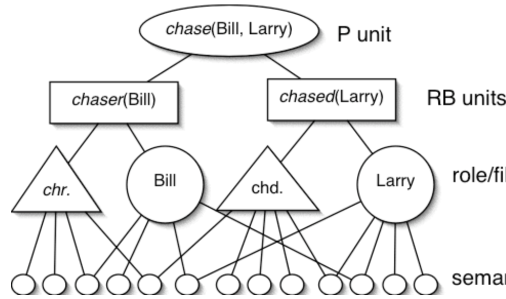

Figure 1. Illustration of a proposition in DORA. Triangles are used to denote roles and circles to denote objects for clarity. In DORA, the same types of units code both roles and fillers.

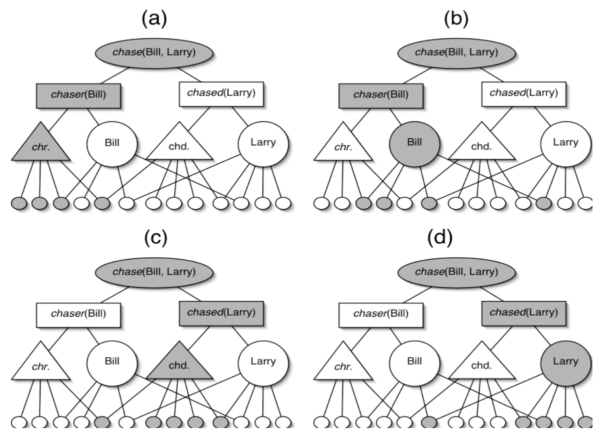

Figure 2. Role-filler binding by asynchrony of firing in DORA. The *chaser* role fires (a), followed by Bill, the object bound to that role (b). Then, the *chased* role fires (c), followed by Larry, the object bound to that role (d).

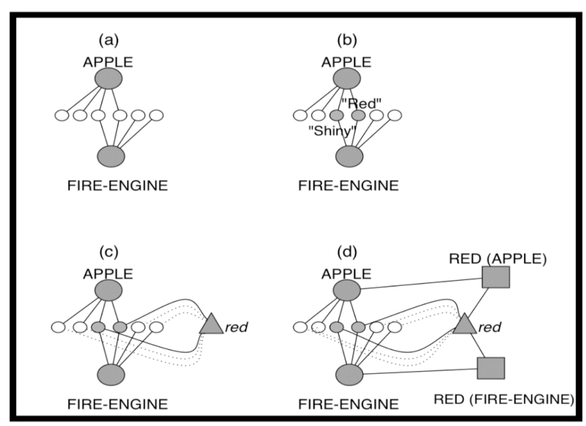

Figure 3. Comparison based predication. (a) An apple and a fire-engine are compared. (b) Semantics shared by both objects become more active than unshared semantics. (c) A new unit is recruited and learns connections to the active semantics in proportion to their activation. (d) The new unit is bound to the apple and to the fire-engine via RB units. Solid lines = stronger connections, dashed lines = weaker connections. Gray = more active units.

active as unshared semantics (Figure 3b). When a pair of mapped roles or fillers fires, DORA recruits a unit to learn connections to the active semantics, where the learned weights are proportional to the semantic units' activations (Figure 3c). The new unit can then be bound to the role/filler units that were compared to create it via an RB unit (Figure 3d). Thus the shared properties of the apple and the fire-engine (here the properties "red" and "shiny") are explicitly predicated about these objects.

We use the example of learning the predicate *red* only because it is easy to explain. Applied to more complex arrangements of objects, the very same operations permit DORA to learn more complex relational roles (e.g., *has-size* (*x*), *more-high-than-something* (*x*), *beside-something* (*x*)) that can be used to construct complex relational representations (e.g., *higher* (*x*, *y*), *beside* (*x*, *y*); see *From Predicated Object Attributes to Whole Relations* and *Simulations* sections, and Doumas et al., in prep.).

Representation Refinement via Schema Induction

Consistent with numerous findings in the developmental literature (for a review see Smith, 1989), new predicates are initially very context dependent. Most compared objects share a number of extraneous features (e.g., the apple and the fire engine were both "shiny" and "red"). To refine predicate representations DORA uses its systematic asynchronous binding routine coupled with an adaptation of LISA's schema induction routine.

When propositions are compared corresponding elements are mapped. If, for example, our child compares two representations of explicitly red objects the *red* predicates will map, as will their fillers (Figure 4a). Because roles fire in direct sequence with their fillers, the mapped predicate representations fire out of synchrony with their fillers. Semantics shared by the two *red* predicates will become twice as active as unshared semantics. Using a simple self-supervised learning algorithm (Hummel & Holyoak, 2003) token units are recruited to match active mapped pairs in the driver and recipient propositions. So, a role/filler unit will be recruited to learn connections to the active semantics (encoding the overlap of the two *red* predicates; Figure 4b). When the mapped objects fire, a second role/filler unit will be recruited to learn connections their shared semantics. In addition, an RB unit will be recruited to encode the binding of the two new role/filler units (Figure 4c). Thus, a refined and schematized representation of the property *red* is formed.

From Predicated Object Attributes to Whole Relations

DORA provides a number of ways to learn whole relational representations (Doumas et al., in prep.). One of the most fundamental involves learning relations by mapping sets of co-occurring role-filler sets. If multiple role-filler sets enter DORA's WM together, it can map them, as a set, onto other co-occuring role-filler sets it has experienced. For example, if DORA had previously experienced that a car was big, and a matchbook small, and then it noticed a train was big and a mouse was small, it could map *big*(train) to *big*(car) and *small*(mouse) to *small*(matchbook). This processes leads to a distinct pattern of firing over the units composing each set of propositions (namely, the RB units of *big*(train) fire out of synchrony with those of *small*(mouse) while the RB units of *big*(car) fire out of synchrony with those of *small*(matchbook)). This pulsing activation over sets of units acts as a signal to link oscillating RB units with a P unit. This process results in a rudimentary (and context dependent) representation of a relation (here *bigger-than* (train, mouse)). Subsequently, the same schema induction routine that serves to refine predicate representations serves to refine whole relation representations, producing context independent representations of whole relations.

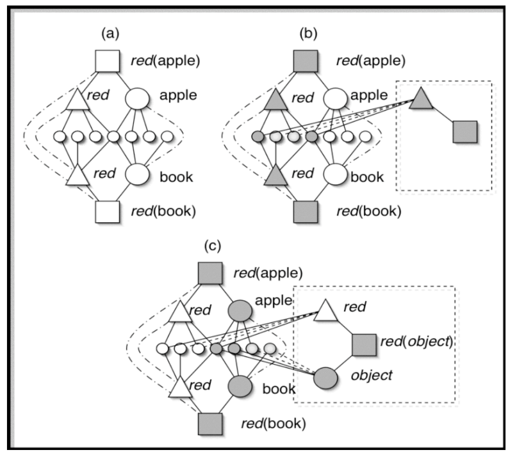

Figure 4. Asynchronous schema based refinement. (a) The *red*+apple role-filler binding maps to the *red*+book role-filler binding (semi-dashed lines indicate mapping connections). (b) The *red* predicates activate their semantics (darker gray indicates more activity). New units are recruited to respond to the active predicates and the active RB (full lines indicate stronger connections, dashed lines weaker connections). (c) The apple and book objects become active, activating their semantics. A new object unit is recruited to respond to the newly active objects.

Simulations

Smith (1984) tested children's ability to match items based on identicality, shared properties, and shared relations. The experimenters each selected two object based on identicality (e.g., both selected two identical items), a shared property (e.g., both selected two red items of different types), or a shared relation (e.g., experimenter 1 chose two red items of different size and shape, and experimenter 2 chose two blue items of different size and shape). The child then had to select two items based on the same rule as that used by the experimenters. They found that two, three, and four-year olds could select items based on identicality, but only three and four-year olds could consistently select items based on shared properties, and only four-year olds could consistently select items based on shared relations.

DORA predicts this exact trajectory. Beginning with holistic object representations, DORA abstracts out and predicates representations of object properties (i.e., single place predicates). It then concatenates sets of single-place predicates to form whole relations. When DORA only represents objects, it can match based on over-all featural similarity (e.g., it can match two red balls because they share many semantic features), like the 2 year-olds in Smith's study. Once DORA has predicated object properties, it can match objects based on their shared properties (e.g., it can match the propositions *red* (*x*) to *red* (*y*) based on the similar predicates), like the 3 year-olds in Smith's study. When DORA learns relations by concatenating single-place predicates it can match objects based on relational similarity (e.g., it can match the propositions *same-color* (*x*, *y*) with *same-color* (*a*, *b*) based on the similar relations), like the 4 year-olds in Smith's study. Importantly, DORA *must* follow this trajectory. DORA learns structured representations of object properties from unstructured representations of objects, and uses them to form representations of relations.

We also used DORA to simulate the findings of Smith, Rattermann, & Sera (1988). In this study children ages 4-5 were presented with pairs of toy butterflies at three different sets of heights: (1) One butterfly at one foot the other at two feet; (2) one at three the other at four feet; (3) one at five and the other at six feet. The child was asked whether one of the two butterflies was higher (or lower) and if so which one. On *consistent* trials both butterflies were high (or low) and the child was asked whether one was higher (or lower). On *neutral* trials both butterflies were in the middle (at three and four feet) and the child was asked whether one was higher or one was lower. On the *inconsistent* trials both butterflies were high (or low) and the child was asked whether one was lower (or higher). The 4 year-olds performed well on the consistent trials, but progressively worse on neutral and inconsistent trials. The 5 year-olds performed well on all trial types (see Table 1).

DORA can learn to appreciate variations in the magnitude of quantifiable properties (i.e., it learns the relation *more* (*x*, *y*); Doumas et al., in prep.), so in this simulation if DORA predicated a quantifiable property (e.g., height) about two objects it was allowed to apply the *more* (*x*, *y*) relation to the role-filler sets (i.e., to represent that one of the two objects had more of that property than the other). We held the representation of the problem constant across "age" in the driver. For all trial types a proposition expressed that one butterfly had more height than the other (*more-height*(butterfly1, butterfly2), see below). For the consistent and inconsistent trial types two additional propositions expressing that butterfly1 and butterfly2 were *high* or *low* (e.g., *high*(butterfly1) and *high*(butterfly2)) were included. We varied the proportion of knowledge in LTM by "age". We simulated four year-olds with 30 value-dependent representations (e.g., *higher* with semantics of "high" and *lower* with semantics of "low") and 10 value-independent representations of height (e.g., representations of height without "high" or "low" semantics). We simulated five-year olds with 30 value-dependent and 30 value-independent representations of height. At both "ages" we also included 30 random propositions about butterflies.

This distribution of propositions in LTM was chosen as it mirrors DORA's learning trajectory. Starting with representations of objects with semantics like as "high" and "low" and "has-height" it learns single place explicit representations of high and low. Applying the *more* relation to pairs of high and low items produces loaded representations of *higher* and *lower* (i.e., *higher* things that are "high" and *lower* things that are "low"). As DORA compares high items to low items it begins to extract less value-laden representations of the abstract property of *height* (i.e., *has-height* (*x*)). Applying the *more* relation to pairs of items with the value-independent representation of height predicated about them produces value free representations of *higher* and *lower* (i.e., *more-height* (*x*, *y*)). What is important here is that DORA *must* learn the value-dependent representations first because it learns value-independent representations from them.

We had DORA retrieve propositions from LTM using a retrieval algorithm based on the LISA model (Hummel & Holyoak, 1997) and then map retrieved propositions onto the propositions in the driver. The proposition that mapped most strongly to a driver proposition was selected as the winner and DORA answered the question which butterfly is higher/lower using this proposition (e.g., if it mapped a representation of higher onto a predicate that took butterfly1 as an argument it answered butterfly1 in response to the question "which is higher" and guessed in response to the question "which is lower"). Our results very closely matched the empirical data (see Table 1). Because DORA is a process model, our interest is with qualitative fits of data. Importantly, the qualitative patterns we report are very robust with respect parameter values.

Table 1.

	Consistent	Neutral	Inconsistent
Children age 4	75.1	73.8	57.5
DORA age 4	67.3	61.4	53.5
Children age 5	87.6	86.5	81.5
DORA age 5	88.1	88.9	82.4

We have also used DORA to simulate a number of other empirical findings (Doumas et al., in prep.).

Discussion

We have presented a theory of predication and relation discovery embodied in DORA, a computational model. DORA provides a systematic account of how object properties and relational concepts can be learned from examples. The primary hypothesis motivating DORA is that the same cognitive operations that exploit relational representations (namely, analogical mapping and schema induction) are fundamental to their discovery. Our account rests on a set of core theoretical claims. First, representing knowledge in a role-filler binding scheme reduces the problem of relational discovery to the problem of predicating object properties and linking them to form full relational structures. This makes the problem of relation discovery tractable. Second, comparison coupled with intersection discovery can bootstrap the predication of object properties. Comparison-based predication requires that roles and objects share a common representational basis, and that the mechanism for binding roles to fillers not only explicitly represents role-filler bindings, but keeps bound roles and fillers distinct for the purposes of processing.

Starting with holistic representations of objects, DORA learns structured representations of object properties (i.e., single-place predicates) and relations (i.e., multi-place predicates). In so doing it provides a natural account of children's progression from holistic to more structured knowledge representations. One limitation of DORA is that currently it does not speak to the question of where the semantic features of objects come from in the first place. (It is worth noting that DORA is not alone in this respect. All computational models are forced to assume some population of primitives.) A complete account of relation discovery should address both how structured representations arise from holistically represented collections of primitive features, as well as the origins of those primitive features. DORA provides a solution to the first of these problems, and we are currently working to generalize the same routines to account for the second.

Acknowledgements

The authors thank Steve Engel, Keith Holyoak, and Cathy Sandhofer for helpful discussions.

References

Doumas, L. A. A., & Hummel, J. E. (in press). Approaches to modeling human mental representations: What works, what doesn't, and why. In K. J. Holyoak & R. Morrison (Eds.), *The Cambridge Handbook of Thinking and Reasoning*. Cambridge: Cambridge University Press.

Doumas, L. A. A., Hummel, J. E., & Sandhofer, C. M. (in prep.). A symobolic-connectionist model of predication and relation discovery.

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

Gasser, M. & Colunga, E. (2001). Learning relational correlations. *International Conference on Cognitive Modeling*, *4*, 91-96.

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

Gentner, D. (2003). Why we're so smart. In D. Gentner and S. Goldin-Meadow (Eds.), *Language in mind: Advances in the study of language and thought* (pp.195-235). Cambridge, MA: MIT Press.

Gentner, D., & Rattermann, M. J. (1991). Language and the career of similarity. In S. A. Gelman & J. P. Byrnes (Eds.), *Perspectives on thought and language: Interrelations in development* (pp. 225 – 277). New York: Cambridge University Press.

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

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

Hummel, J. E., & Holyoak, K. J. (2003). A symbolic-connectionist theory of relational inference and generalization. *Psychological Review*.

Kotovsky, L., & Gentner, D. (1996). Comparison and categorization in the development of relational similarity. *Child Development*, 67, 2797-2822.

Love, B. C. (1999). Utilizing time: Asynchronous binding. *Advances in Neural Information Processing Systems*, *11*, 38-44.

Namy, L. L., & Gentner, D. (2002). Making a silk purse out of two sow's ears: Children's use of comparison in category learning. *Journal of Experimental Psychology: General*, 131, 5-15.

Ratterman, M. J., & Gentner, D. (1998). More evidence for a relational shift in the development of analogy: Children's performance on a causal-mapping task. *Cognitive Development*, 13, 453-478.

Sandhofer, C. M, Smith, L. B. (2001). Why children learn color and size words so differently: Evidence from adults' learning of artificial terms. *Journal of Experimental Psychology: General, 130*, 600-617.

Smith, L. B. (1984). Young children's understanding of attributes and dimensions. *Child Development*, *55*, 363-380.

Smith, L. B. (1989). From global similarities to kinds of similarities: The construction of dimensions in development. In S. Vosniadou and A. Ortoney (Eds.), *Similarity and analogical reasoning* (pp. 147-177). Cambridge: Cambridge University Press.

Smith, L. B., Rattermann, M. J., & Sera, M. (1988). "Higher" and "lower": Comaprative interpretations by children. *Cognitive Development*, *3*, 341-357.

Thompson, R. K. R., & Oden, D. L. (2000). Categorical perception and conceptual judgments by nonhuman primates: The paleological monkey and the analogical ape. *Cognitive Science, 24*, 363-396.

von der Malsburg, C. (1981/1994). The correlation theory of brain function. In E. Domany, J. L. van Hemmen, & K. Schulten (Eds.), *Models of neural networks II* (pp. 95-119). Berlin: Springer.

Yamauchi, T., & Markman, A. B. (1998). Category learning by inference and classification. *Journal of Memory & Language, 39*,124-148.