Combining progressive alignment and near-misses to learn concepts from sketches

Matthew D. McLure (mclure@u.northwestern.edu)
Scott E. Friedman (friedman@northwestern.edu)
Kenneth D. Forbus (forbus@northwestern.edu)
Qualitative Reasoning Group, Northwestern University, 2133 Sheridan Rd
Evanston, IL 60208 USA

Abstract

Learning to classify examples as concepts is an important challenge for cognitive science. In cognitive psychology analogical generalization, i.e., abstracting the common structure of highly similar examples, has been shown to lead to rapid learning. In AI, providing very similar negative examples (*near-misses*) has been shown to accelerate learning. This paper describes a model of concept learning that combines these two ideas. SAGE, which models analogical generalization, is used to implement progressive alignment. Near-miss analysis is modeled by using the Structure Mapping Engine to hypothesize classification criteria based on differences between examples with conflicting classifications. Analogical retrieval is used to pull near-misses from memory automatically. We use a sketch understanding system to generate qualitative spatial representations of examples from a corpus of sketched concepts. We show that the model can learn the typical features and classification criteria for a concept from sketches, and that the incorporation of near-miss analysis improves classification performance over progressive alignment and similarity alone.

Keywords: Concept learning; analogy; generalization.

Introduction

How concepts can be learned from experience is a common question for AI systems. It is well known that some concepts can be viewed as analytic, having compact necessary and sufficient defining criteria (e.g., *grandparent* or *triangle)*, whereas others are based on similarity or typicality (e.g., *chair*, *bachelor*). Prior work has explored analogical generalization as an explanation for learning similarity-based categories. The SAGE model of analogical generalization, an evolutionary improvement over SEQL (Kuehne *et al* 2000a) has been used to model learning of perceptual stimuli (Kuehne *et al* 2000b), stories (Kuehne *et al* 2000a), spatial prepositions (Lockwood *et al* 2008) and causal models (Friedman & Forbus, 2008; Friedman & Forbus, 2009). SAGE's ability to construct probabilistic generalizations provides a model of typicality, i.e., high-probability relationships and attributes are more typical. SAGE has been used to model *progressive alignment* (Gentner *et al* 2007), where sequences of highly similar exemplars lead to more rapid learning (Kuehne *et al* 2000a). Progressive alignment alone may suffice to generate rule-like concepts (e.g., Gentner & Medina, 1998), but another possibility is to use negative examples to sharpen criteria for concept membership. Winston (1970) proposed the idea of a *near-miss*, a labeled negative example that differs from the intended concept in a small number of ways. A near-miss exemplar should be highly alignable with some instances of a concept[1]. Consequently, a concept learner that uses similarity alone might misclassify a near-miss negative example as a positive instance, so the similarity can be used as a cue for additional analysis. Presumably, one or more of the differences detected could be critical to the definition of the concept.

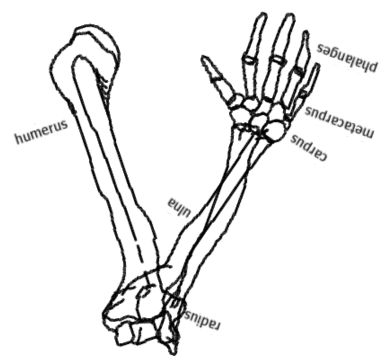


Figure 1: An example of the *skeletal arm* concept, drawn in CogSketch.

Our system uses sketched examples as inputs for learning, which are automatically processed into qualitative spatial representations by a sketch understanding system. Representing the examples with structured, qualitative spatial relations allows for the system to use three analogical processes for learning: generalization via SAGE, analogical retrieval of near-miss examples, and near-miss analysis via analogical inference. The generalizations and classification criteria produced by these processes use the same vocabulary of qualitative spatial relations and functional relationships, which we believe provides a concise, human-like representation system.

This paper describes a model of concept learning that combines analogical generalization and near-miss analysis to capture both similarity-based and analytic aspects of concepts. Its inputs are sketches of physical objects, which are labeled as positive or negative examples of concepts (Figure 1). It uses SAGE to construct generalizations for each concept, thus capturing similarity-based aspects of concepts (and typicality, via probability). When a positive

[1] For disjunctive concepts, some exemplars will not be similar.

Matthew McLure, Scott E. Friedman, Kenneth D. Forbus (2010). Combining progressive alignment and near-misses to learn concepts from sketches. Proceedings of the 24th International Workshop on Qualitative Reasoning. Portland, OR.

example is provided, the corresponding concept generalization(s) are updated, and analogical retrieval is used to find near-misses and generate hypotheses about classification criteria for the concept. Similarly, when a negative example is provided, analogical retrieval is used to find similar prior positive examples. (Using analogical retrieval to find near-misses is a significant advance over Winston's model, which used a single description for concepts and was not capable of finding near-miss examples automatically.)

We show that the model can indeed learn concepts from sketches, and that including near-miss analysis improves learning. Our simulation is implemented using the Companions cognitive architecture (Forbus et al, 2009), which integrates analogical processing and sketching.

The next section summarizes the simulations of analogical processing and sketch understanding upon which our model is built. We then describe our model, the experiment used to test its learning capability, and results. We close with related and future work.

Simulation Components

Analogical processing

Our system uses three cognitive models as components to learn concepts and categorize examples: (1) similarity-based retrieval is used to find similar examples across conceptual boundaries; (2) analogical comparison is used to compare examples and generate classification hypotheses; and (3) analogical generalization is used to generalize examples. We use the Structure Mapping Engine (SME) (Falkenhainer et al, 1989) to model analogical matching, MAC/FAC (Forbus et al, 1995) to model retrieval, and SAGE (Kuehne et al, 2000) to model analogical generalization.

SME is based on Gentner's (1983) structure-mapping theory of analogy. Given two relational representations, a base and a target, SME computes *mappings* that represent how they can be aligned. A mapping consists of correspondences that describe, "what goes with what" in the two representations and a numerical score indicating their degree of similarity. SME also computes *candidate inferences* from the base to the target and from the target to the base. Candidate inferences suggest possible relations that can be transferred across representations, using the correspondences in the mapping as support.

Given a probe case and case library, MAC/FAC efficiently retrieves a case from the case library that is similar to the probe. For scalability, its first stage estimates similarity via dot products on vectors automatically produced from the structured, relational representations used as cases. At most three descriptions are passed to the second stage, which uses SME to compare their full relational versions to the probe, in parallel, to find the best case, or up to three cases if they are very close to the best.

Our model uses SAGE for generalization. Each concept has its own *generalization context*, which SAGE uses to maintain a list of generalizations and ungeneralized examples. Given a new example, it is first compared against each generalization in the context, using SME. If the SME similarity score is over the *assimilation threshold*, the example is merged to update the generalization. Otherwise, the new example is compared with the ungeneralized examples in the context. Again, if the score is over threshold, the two examples are then combined to form a new generalization in the context. Otherwise, the example is added to the context's list of ungeneralized examples. Figure 2 depicts generalization contexts for concepts *Arch* and *Triangle*.

CogSketch

CogSketch[2] (Forbus et al, 2008) is an open-domain sketch understanding system. The ink that a user draws to represent an entity is called a *glyph*, which can be labeled with concepts from an OpenCyc[3]-derived knowledge base. For example, in the sketch shown in Figure 1, each bone is labeled a `Bone-BodyPart`, which is stored as an attribute for each of the individual entities.

CogSketch automatically computes qualitative spatial relations (e.g., `above`, `rightOf`, `touchesDirectly`) between glyphs. In the knowledge representation that is produced by CogSketch, these relations are automatically applied to the entities that the glyphs represent. The abstraction provided by qualitative representations greatly facilitates learning via structural alignment, since an entire space of quantitatively similar configurations will lead to the same qualitative representation.

CogSketch also computes candidate *visual/conceptual relations* (again, from the OpenCyc-derived knowledge base) for pairs of sketched entities based on the visual relationships that hold between them the conceptual labels they have been assigned, and the genre and pose of the sketch. For example, the fact that the glyphs depicting the carpus and metacarpus in Figure 1 touch suggests that the objects they depict might be touching or connected in some way. The list of candidate visual/conceptual relations for these objects is further constrained by the `Bone-BodyPart` concept labels they have been assigned, as well as the `Physical` genre and `from-side` pose of the sketch. The

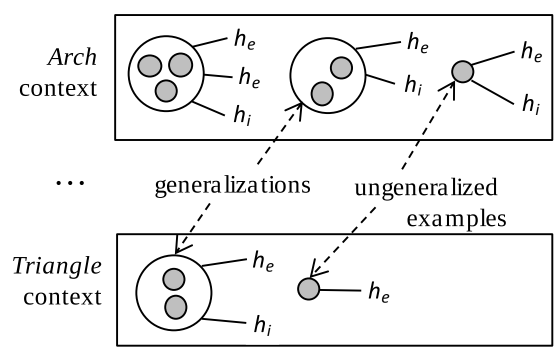


Figure 2: SAGE generalization contexts for *Arch* and *Triangle* concepts, with inclusion and exclusion hypotheses.

[2] http://www.qrg.northwestern.edu/software/cogsketch/
[3] www.opencyc.org

user can browse the candidate relationships and select those that are accurate.

CogSketch is based on the observation that people talk when they sketch, providing verbal labels for what they are drawing, and using language to express functional relationships (e.g. that two parts can rotate, or that one supports another) that the sketch alone cannot convey. The conceptual labels described above, which are applied by a simple menu system, model the effect of verbal labeling. The possible visual/conceptual relationships described above, which are computed automatically and are available for the user to choose or not, model the effect of providing functional information via language. This makes the input process much closer to what happens in human-to-human sketching. The user draws ink, which CogSketch's visual system analyzes, producing visual and spatial relationships. The user-supplied conceptual labels plus the visual/spatial analysis enables CogSketch to automatically compute visual/conceptual relationship candidates, from which the user can select, if they choose. (In the experiments reported here, correct visual/conceptual relationships were always chosen, thereby providing some functional information about the concept.)

Similarity & near-miss concept learning

Our model takes as input a stream of labeled sketches, which have been encoded into qualitative propositional representations by CogSketch. A positive label indicates that the example is an instance of a concept (e.g. an arch).

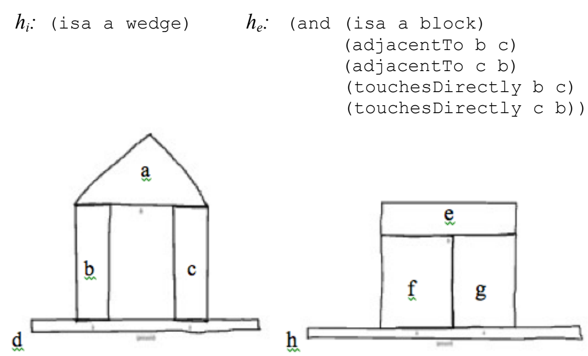


Figure 3: A near miss of concept *arch* and the resulting inclusion hypothesis h_i and exclusion hypothesis h_e.

A negative label indicates that, whatever it is, it is not an example of that concept (e.g. not an arch). Currently the model assumes that concepts are mutually exclusive. When the first positive training example for a new concept is provided, a generalization context is created for that concept. Positive examples are added to the appropriate generalization context, invoking SAGE on it. MAC/FAC is used to find a negative example similar to the positive example. If a sufficiently similar example of a different concept is found, near-miss analysis is invoked. Similarly, when a negative example is provided, MAC/FAC is used to retrieve the closest positive example for near-miss analysis.

When given an example to categorize, the model uses MAC/FAC to generate a reminding from each concept's context. The system tests the new example against the classification criteria for each concept. Of the concepts whose criteria are satisfied, the one with the most similar reminding is chosen as the category of the new example.

In explaining our model, we use as a running example learning the concept of an *arch*, which was first used by Winston (1970).

Near-miss analysis. Winston argued for the importance of *near misses* in learning concepts. A near miss consists of a positive example e_1 (e.g. Figure 3, left) and a negative example e_2 (e.g. Figure 3, right) that differ only slightly in their structured representations. In analogical reasoning terms, e_1 and e_2 are highly alignable, enabling a learner to conjecture that differences between them could be criteria for classification. Two kinds of hypotheses are computed. *Inclusion hypotheses* represent potential necessary conditions for something to be an instance of the concept. *Exclusion hypotheses* represent potential negative conditions that are sufficient to prevent positive classification. All hypotheses are reified as structural relational expressions.

Near-miss analysis starts with a positive and a negative example. As noted above, one of these examples is a new training example, while the other is a previous example retrieved via MAC/FAC. A similarity threshold of 0.75 is used for their comparison, to ensure high alignability.

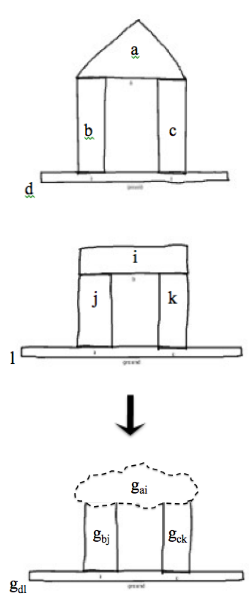
Figure 3 shows a near-miss that was processed by our simulation. The positive example is used as the base, the negative example as the target, and they are compared via SME. SME aligns *a* with *e*, *b* with *f*, *c* with *g*, and the grounds *d* with *h*. The conjunction of positive→negative candidate inferences in the mapping becomes a new inclusion hypothesis (Figure 3, h_i) designating criteria that might be necessary for concept membership. Similarly, the conjunction of all negative→positive candidate inferences becomes a new exclusion hypothesis (Figure 3, h_e) designating criteria that might prevent concept membership. Here the attribute `(isa a wedge)` is the sole forward candidate inference, so it becomes the inclusion hypothesis h_i. This hypothesis posits that in any unclassified example, the entity alignable with *a* must have the `wedge` attribute to qualify the example as an arch. Similarly, the `block` attribute, `touchesDirectly` relations, and `adjacentTo` relations comprise the conjunctive exclusion hypothesis h_e.

Inclusion and exclusion hypotheses are associated with the positive example in the near miss that generated them, as shown in Figure 2. Consequently, when MAC/FAC retrieves more than one near-miss for a given positive example, the system computes a corresponding number of hypothesis about the example, and must reconcile them. Inclusion hypotheses pertaining to the same example are combined via set union, since all necessary facts must hold for positive classification. Conversely, any exclusion hypothesis suffices to rule out that concept, so they are kept separate.

In Figure 3, the inclusion hypothesis h_i generated by the system erroneously asserts that all arches have wedges as their top. This error reflects one learning bias of the model, which is the immediate assumption that all differences detected in the near miss of a concept are important to the definition of the concept. Such errors can be removed during analogical generalization, which we discuss next.

Analogical generalization. During training, our learning system incrementally develops a disjunctive model of a concept through the observation of positive and negative examples. As positive examples are observed, they are added to a SAGE generalization context for the concept, where they are generalized with sufficiently similar examples. When an example is generalized, resulting in new or larger generalizations (shown in Figure 2) the system revises the hypotheses associated with the generalization.

Across generalizations, the near-miss hypotheses can be considered disjunctive hypotheses about the concept. For example, suspension bridges may be different enough from beam bridges that the classification hypotheses required of them differ. We can capture this distinction if suspension bridge examples and beam bridge examples form separate generalizations when added to the generalization context for the concept *bridge*. During classification, we may claim that an example is a bridge if it is similar enough to the *suspension bridge* generalization and satisfies the qualitative conditions for *suspension bridge*, or if it is similar enough to the *beam bridge* generalization and satisfies the qualitative conditions for *beam bridge*. The construction of disjunctive hypotheses based on similarity introduces another learning bias of the model, which assumes that similar examples of a concept are subject to the same rules for membership.



h_i: (isa a Wedge)
h_e: (and (isa a Block)
(adjacentTo b c)
(adjacentTo c b)
(touchesDirectly b c)
(touchesDirectly c b)

h_i: (and (isa i Block)
(onPhysical i k)
(touchesDirectly i k)
h_{e1}: (and (adjacentTo j k)
(adjacentTo k j)
(touchesDirectly j k)
(touchesDirectly k j)
h_{e2}: (isa i Wedge)

h_i: (and (onPhysical g_{ai} g_{ck})
(touchesDirectly g_{ai} g_{ck}))
h_e: (and (adjacentTo g_{bj} g_{ck})
(adjacentTo g_{ck} g_{bj})
(touchesDirectly g_{bj} g_{ck})
(touchesDirectly g_{ck} g_{bj}))

Figure 4: The generalization of two positive examples and their inclusion and exclusion hypotheses.

After an observed positive example is generalized with an existing generalization or ungeneralized example, their hypotheses are generalized. Figure 4 shows how a new example (top) and a previously ungeneralized example (middle) are merged into a new generalization with revised hypotheses (bottom).

The first step in generalizing inclusion hypotheses is mapping the hypotheses from their respective generalized examples to the newly created generalization. This involves replacing the names of entities with the names of corresponding entities in the generalization. Next, inclusion hypotheses are pruned by removing any assertions that do not hold on the new generalization. In Figure 4, the facts `(isa a Wedge)` and `(isa i Block)` are pruned from the inclusion hypotheses of the constituent examples because they are not true of the resulting generalization, i.e., the corresponding generalized entity g_{ai} is not known to be either `Wedge` or `Block`. After pruning, the facts of the two inclusion hypotheses are unioned to create a conjunctive hypothesis associated with the new generalization

Next, the system uses the generalization operation to identify and discard erroneous exclusion hypotheses. In Figure 4, exclusion hypothesis `(isa i Wedge)` of the middle example is erroneous because it shares a generalization with the topmost example whose corresponding entity *a* is a `Wedge`. Consequently, the exclusion hypothesis is discarded. Remaining exclusion hypotheses are mapped onto the resulting generalization. Finally, the system discards exclusion hypotheses of the resulting generalization that are more specific than other associated hypotheses (i.e., for every exclusion hypothesis composed of fact set *f*, any hypothesis of fact set *f'* such that $f \subseteq f'$ is eliminated). In Figure 4, hypothesis h_e of the topmost example is discarded for this reason.

Classification

Given a new testing example e_{new}, our model decides whether it is an instance of one of its learned concepts. The model decides this using similarity-based retrieval and by testing the hypotheses created during learning.

For each learned concept, the system uses MAC/FAC to retrieve the most similar generalization or ungeneralized example of the concept e_c from the concept's generalization context. The inclusion and exclusion hypotheses associated with e_c (as shown in Figure 2) are used as criteria for classifying e_{new}.

The inclusion and exclusion hypotheses associated with e_c are represented in terms of the entities in e_c, which typically do not exist in e_{new}. Consequently, structural alignment is used to perform the analogical equivalent of rule application. SME is used to find entity correspondences between e_c and e_{new}, and the entities of e_c are substituted with the corresponding entities in e_{new} in each hypothesis.

Testing the classification criteria is the final step in classification. If an inclusion hypothesis does not hold in e_{new}, or if an exclusion hypothesis does hold in e_{new}, it is not an instance of the concept. Otherwise, e_{new} is an instance of the concept. If e_{new} is a viable instance of multiple concepts, given the exclusion and inclusion criteria, the system chooses the concept whose MAC/FAC reminding similarity score was higher. Thus our model of concepts combines both rule-based and similarity-based aspects.

Experiment

We created a series of 44 sketches representing six concepts for learning and categorization, summarized in Table 1. The *false arches*, *false triangles*, and *false squares* sketches are all highly alignable with examples of their associated concept, but are not positive examples themselves.

Table 1: Sketched examples for simulation.

Arches:	8	Triangles:	4
False arches:	8	*False triangles:*	4
Bridges:	4	Squares	4
Skeletal arms:	4	*False squares:*	4
Skeletal legs:	4		

Our experiment follows a four-fold cross validation format covering all 44 sketches. The sketches were randomly assigned to four groups (folds) of 11 sketches each, with the constraint that all groups had the same distribution of sketches from the categories in Table 1 (two arches, two false arches, one bridge, one skeletal arm, etc). The system trained on three 11-example groups, for a total of 33 examples for learning. The remaining group of 11 examples is used for classification testing. We repeat this four times, so each group of 11 examples is used once for testing, resulting in 44 classifications total.

We tested our simulation under two conditions: The *full* condition uses the entire model, while in the *similarity-only* condition, near-miss analysis is turned off. In similarity-only, the system classifies a new example by using MAC/FAC to retrieve a similar representation from the concept context, and asserts concept membership if the normalized SME similarity score is above a threshold of 0.85. We expected that, based on prior experiments (Kuehne *et al* 2000b), similarity-only will learn quite well with only a handful of examples. However, we also expect that misleadingly similar negative examples will cause false positives, which near-miss analysis should help prevent.

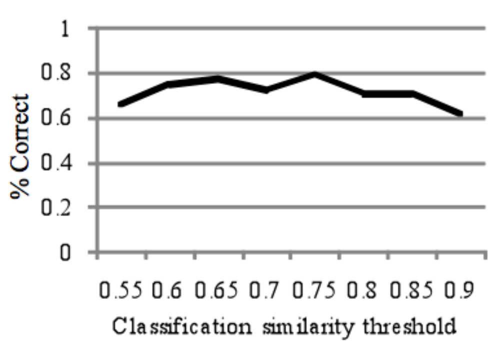


Figure 5: Effectiveness of using structural similarity alone for classification, as a function of similarity threshold.

In the similarity-only condition, 79% correct classification is achieved with a similarity threshold of 0.75 (Figure 5), well above chance ($p < 0.001$). Inspection of the results revealed that almost all of the 20% error can be attributed to false positives. One such false positive is the rightmost example in Figure 3, which shares considerable relational structure with other arches.

With near-miss analysis turned on, 86% correct classification was achieved, which is better than chance with $p < 0.001$. The number of false positives decreased from eight to two but the number of false negatives increased from one to four due to overly restrictive hypotheses. The rightmost example in Figure 3 was among the negative examples correctly classified. Just as with similarity-only, the model determined that this example was sufficiently similar to a generalization of the concept *arch*. However, it reported a failure to meet classification conditions due to a satisfied exclusion hypothesis,

```
(TheSet (adjacentTo f g)
        (touchesDirectly g f))
```

which expresses the justification "This is not an arch because f is adjacent to g and g touches f directly."

Discussion & Future Work

We have described a model that extends analogical generalization with near-miss analysis to learn concepts from sketches. We have generalized the notion of near-miss that Winston (1970) used in two important ways. First, Winston assumed that near-misses were always provided by a teacher. We have shown that near misses can also naturally arise from the process of similarity-based retrieval, thereby providing more self-direction in learning. Second, Winston's system had one description of the target concept it was learning, and hence did not capture the possibility of disjunctive concepts and finding the appropriate conceptual representation, which we do via a combination of SAGE and MAC/FAC. A version of the model without near-misses, using similarity alone, performs well over chance. However, similarity alone leads to a pattern of misclassification errors, which is partially corrected by near-miss analysis. The incorporation of qualitative classification criteria enables the model to make more expressive justifications for its classification decisions, as in the case of the negative example from Figure 3. We also believe that near-miss analysis will allow the model to more readily benefit from a larger training set, as hypotheses from new near-misses will add potentially valuable criteria to reduce false positives and hypothesis generalization will alleviate over-restrictiveness, which accounted for all but one of the false negatives. We expect the similarity-only classifier to gain less from additional training, since the examples it misclassifies are mostly negative examples that bear high relational similarity to positive examples. Thus near-miss analysis provides an important extension to similarity-based concept learning.

Our concept learning model learns several concepts simultaneously, with relatively few examples. It requires orders of magnitude fewer examples than existing connectionist models of concept learning (e.g., Krushke, 1992; Regier 1996; Elman 1999), and unlike such models, uses automatically encoded qualitative representations as stimuli, to reduce tailorability. We believe our model makes more realistic demands, although it could be argued that our model learns too quickly. One reason that we see such rapid learning in simulation experiments is that our system, unlike people, has many fewer distracters. Everyday life does not always afford closely packed sequences of similar concept instances, and human perception may contain more attributes and relations than CogSketch currently computes. However studies such Gentner *et al* (2009) suggest that people can learn spatial concepts quickly with highly alignable near-misses, which our model captures nicely.

Winston's 1970 system used line drawings of 3D Blocks World scenes as input, which were automatically processed via scene analysis. Our sketch understanding system uses a more general set of 2D representations and hand-drawn sketches, which are noisier than line drawings. Winston (1982, 1986) also explored learning rules from analogies, using simplified English inputs. His system generalized based on one example, rather than several, and produced logical quantified rules, while ours uses analogical matching to apply qualitative hypotheses to new examples. His if-then rules and censors are functionally similar to our inclusion and exclusion hypotheses, respectively.

There are several aspects of concept learning that our model does not currently capture. For example, our sketched input does not include causal relationships or goals (Lombrozo, 2009; Rehder & Kim, 2006). Based on prior work (Falkenhainer, 1987; Friedman & Forbus, 2009) we believe our model will handle such information if it is included in the initial encoding, since it basically adds relational structure that influences similarity judgments, and hence classification, in our model. Other factors, such as ontological structure (Medin & Smith, 1984), we believe can be handled by further exploiting the statistical information gathered via SAGE in cross-concept analyses. We plan to explore both of these issues in future work.

Acknowledgments

This work is supported by the Cognitive Science Program of the Office of Naval Research.

References

Elman, J. (1999). Generalization, rules, and neural networks: A simulation of Marcus et. al, (1999). Ms., University of California, San Diego.

Falkenhainer, B., Forbus, K. and Gentner, D. (1989). The Structure Mapping Engine: Algorithm and examples. *Artificial Intelligence*, 41, 1-63.

Forbus, K., Klenk, M., and Hinrichs, T. (2009). Companion Cognitive Systems: Design Goals and Lessons Learned So Far. *IEEE Intelligent Systems*, vol. 24, no. 4, pp. 36-46, July/August.

Forbus, K., Lovett, A., Lockwood, K., Wetzel, J., Matuk, C., Jee, B., and Usher, J. (2008). CogSketch. *Proceedings of AAAI 2008*.

Forbus, K., Gentner, D. and Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, 19(2), 141-205.

Friedman, S. & Forbus, K. (2008). Learning Causal Models via Progressive Alignment & Qualitative Modeling: A Simulation. In *Proceedings of the 30th Annual Conference of the Cognitive Science Society*.

Friedman, S. & Forbus, K. (2009). Learning Naïve Physics Models & Misconceptions. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society*.

Gentner, D. (1983). Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science*, 7: 155-170.

Gentner, D., Levine, S., Dhillon, S. & Poltermann, A. (2009). Using structural alignment to facilitate learning of spatial concepts in an informal setting. In *Proceedings of the Second International Workshop on Analogy*, Sofia, Bulgaria, 2009.

Gentner, D., Loewenstein, J., & Hung, B. (2007). Comparison facilitates children's learning of names for parts. *Journal of Cognition and Development*, 8. 285-307.

Gentner, D. & Medina, J. (1998). Similarity and the development of rules. *Cognition* 65(2-3):263-97.

Kruschke, JK (1992). ALCOVE: An exemplar-based connectionist model of category learning. *Psychological Review 99*, 22-44.

Kuehne, S., Forbus, K., Gentner, D. and Quinn, B. (2000). SAGE: Category learning as progressive abstraction using structure mapping. *Proceedings of CogSci 2000*.

Kuehne, S., Gentner, D. and Forbus, K. (2000). Modeling infant learning via symbolic structural alignment. *Proceedings of CogSci 2000*.

Lockwood, K., Lovett, A., and Forbus, K. (2008). Automatic Classification of Containment and Support Spatial Relations in English and Dutch. In the *Proceedings of Spatial Cognition 2008*.

Lombrozo, T. (2009). Explanation and categorization: how "why?" informs "what?". *Cognition*, 110, 248-253.

Medin, D. and Smith, E. (1984). Concepts and concept formation. *Annual Reviews of Psychology*, 35, 113-138.

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

Rehder, B. & Kim, S. (2006). How causal knowledge affects classification: A generative theory of categorization. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32, 659-683.

Winston, P.H. 1970. Learning structural descriptions by examples. Ph.D. thesis, MIT.

Winston, P.H. 1982. Learning new principles from precedents and exercises. *Artificial Intelligence* 23(12).

Winston, P.H. 1986. Learning by augmenting rules and accumulating censors. In Michalski, R., Carbonell, J. and Mitchell, T. (Eds.) *Machine Learning: An Artificial Intelligence Approach, Volume 2.* Pp. 45-62. Morgan-Kaufman.