Enhancing Methodological Rigor for Computational Cognitive Science: Complexity Analysis

Jacob Beal (jakebeal@bbn.com)
BBN Technologies, 10 Moulton Street
Cambridge, MA 02138 USA

Jennifer Roberts (jenmarie@mit.edu)
MIT CSAIL, 32 Vassar Street
Cambridge, MA 02139 USA

Abstract

Complexity analysis provides a measure of how well the computations being performed by a cognitive model are supported by the constraints of biological computing. We argue that research in computational cognitive science will be greatly aided by treating computational complexity as a primary issue on the same order of importance as significance analysis in experimental work. In this paper, we give a brief review of computational complexity and its application in computer science. We then show how it can be applied to incorporate biological constraints abstractly into computational cognitive science research, even in the presence of massive uncertainty about the underlying neuroscience. To ground this discussion of complexity, examples are drawn from two recent pieces of work: Xu and Tenenbaum's work applying Bayesian inference to word learning (Xu & Tenenbaum, 2007) and Forbus and Hinrich's work on the construction of a large-scale analogical reasoner (K. D. Forbus & Hinrichs, 2006).

Keywords: Methodology; Asymptotic Complexity; Complexity Analysis; Biological Limitations

Motivation

Cognitive scientists often produce computational models that simulate the operation of cognitive processes. The discussion of a model, however, commonly neglects its computational complexity. Cognitive science thus abandons one of the most fundamental tools computer science provides for the analysis of models. This impairs our ability to clearly compare competing models and thereby encourages the development of both models that work on particular problems but cannot extend to broader problems and models that are likely to be rendered obsolete by small changes in our understanding of the biology of the brain.

Complexity analysis helps address such problems by providing a measure of how well the computations being performed by a model are supported by the constraints of biological computing. We argue that research in the field will be greatly aided by treating computational complexity as a primary issue on the same order of importance as significance analysis in experimental work, rather than an optional property that is useful when it can be established.

In this paper, we give a brief review of computational complexity and its application in computer science. We then show how it can be applied to incorporate biological constraints abstractly into computational cognitive science research, even in the presence of massive uncertainty about the underlying neuroscience. To ground this discussion of complexity, we use examples from two recent pieces of work: Xu and Tenenbaum's work applying Bayesian inference to word learning (Xu & Tenenbaum, 2007) and Forbus and Hinrich's work on the construction of a large-scale analogical reasoner (K. D. Forbus & Hinrichs, 2006). We have chosen these two pieces of solid, well-formulated computational work to illustrate how complexity analysis can enhance qualitatively different types of computational research.

A brief review of computational complexity

Let us start with a brief review of computational complexity, which should be familiar to anyone with a computer science background. When computer scientists speak of "computational complexity," what they are generally referring to is *asymptotic* computational complexity—how the resources consumed by a model grow as its scale increases. This resource consumption can stem from costs incurred by the data structure used to represent the model (e.g. the number of nodes and edges in a Bayesian network) or by the algorithms that implement the model's operations (e.g. performing inference in a Bayesian network) or by the algorithms necessary for modifying the model (e.g. changing a conditional probability table in a Bayesian network).

If we use the positive number n to describe the scale of a model, then the model has upper bound asymptotic complexity $O(f(n))$ if and only if there are constants n' and k such that whenever $n > n'$, the resource cost of the model is less than $kf(n)$. For example, a Bayesian network with n nodes can have up to $\sum_{i=1}^{n}(i-1)$ dependencies, so it takes $O(n^2)$ space to represent its graphical structure. A node in the network with p parents has a conditional probability table with 2^p entries, requiring $O(2^p)$ space. If the greatest number of parents of any node in the network is P, then the cost of storing the entire network—nodes, dependencies, and conditional probability tables—is $O(n2^P)$.

Although people most frequently use the "Big O" bound, other complexity bounds are important as well, such as the Ω asymptotic lower bound and the Θ asymptotic tight bound. People most often analyze costs for time and space, and while they are usually focus on the worst case cost, we may also be interested in the expected cost or the amortized cost.[1]

[1] Amortized cost shows how an expensive operation can be "paid for" by improved performance on many other operations. For exam-

When possible, complexity bounds are established through formal analysis. For many models, however, this is either not practical or yields bounds that are too loose. This need not stop us, any more than it has stopped digital electronics, where metastability theorems show formally that a digital circuit might never converge, and yet every execution of a modern processor does so in approximately one nanosecond. In these circumstances, one can often characterize the scaling of a model through careful experiment, establishing an operating range in which the behavior of the model is well understood and making projections outside of this range in areas where no unmeasured phenomena are expected to intrude. For example, graph search is $O(b^d)$ for branching factor b and depth d to the goal, but if heuristic search through a highly porous planar graph like a road network were determined empirically to be, say, $O(d)$ for several cities, then it would be reasonable to predict that it would hold for nationwide road networks, because the character of the network is not particularly different at the larger scale.

Why care about computational complexity?

In cognitive science, we study of the nature of intelligence, but even the most ambitious of today's computational models only address a small fraction of the whole. Therefore, every computational model must aim to advance us toward the Holy Grail of the subject, a "broad model" of intelligence that addresses the entirety of a mind.

How a model maps onto that goal may vary—some aim to be used as elements of a broad model (e.g. feature maps for the V1 brain area), some aim to describe abstractions implemented somehow by a broad model (e.g. Jackendoff's semantic models), and some, most indirectly, aim to identify computations that must at least be approximated by a broad model (e.g. Tenenbaum's program of Bayesian reasoning).

In every case, however, proposing a computational model is also proposing a computation that the brain must be able to perform using its sharply limited computational resources. Remember, the entire human brain contains only an estimated 10^{11} neurons (Williams & Herrup, 1988) and an estimated 10^{15} synapses (Shepherd, 2004), and, taking the time scale on which a signal passes along a neuron to be about one millisecond, a human lifetime contains only about 10^{13} "clock cycles" of computation.[2] While these numbers are vast compared to, say, the number of words in a person's vocabulary, they fall quickly before even moderate complexity.

Using these order-of-magnitude estimates, we can make a three way comparison between the resources available in a brain, the complexity of a model, and the scale of the phenomena being modelled. Combining the uncertainty in our knowledge about brain structure, how brains compute, how a particular model might be implemented by a brain, and the scale of the phenomena being modelled (vocabulary, for instance, debatably contains anywhere from 10,000 to 1 million known words, depending on the definition of "word" and "know"), we can expect several orders of magnitude uncertainty in our estimates. Even so, we can readily judge the plausibility of a model.

For example, taking the highly conservative estimate of 10,000 words in an adult vocabulary, a model of language understanding that used $O(w) = k \cdot 10^4$ space would be beautifully cheap and one that cost $O(w^2) = k \cdot 10^8$ space would be reasonable. But $O(w^3) = k \cdot 10^{12}$ is starting to devote an awful lot of space to words, and $O(w^4) = k \cdot 10^{16}$ is unreasonable unless one can pack many bits per synapse or ensure that the constant will be much less than one. Higher complexities, like $O(w^{10})$ or $O(2^w)$, are simply ridiculous.

Figure 1 shows some other examples of reasonable complexity bounds implied by the relationship between available biological resources and a modeling domain. Even though the particular numbers are highly debatable, the uncertainty makes little difference to the complexity bounds. For example, consider how vocabulary complexity bounds would be affected if our model of the human brain shifts so that glial cells are designated as important computational elements and their number is estimated at 50 times the number of neurons. This raises the number of computational elements from 10^{11} to 10^{13}—a hundred-fold increase that at first appears massive. But the cube root of 100 is less than 5, and so while this change makes it somewhat easier to argue for the plausibility of a model that requires $O(w^3)$ space, it by no means assures it. It does even less for $O(w^4)$, and nothing at all for the plausibility of higher complexities. Thus, complexity analysis provides guidance that will likely remain valid even through major shifts in our understanding of neuroscience.

The mere fact that a model is unreasonably expensive need not force us to discard it. If the model is attractive for other reasons, such as elegance or compliance with experiment, then we may pursue one of several paths to resolve this difficulty:

- Can components of the model be replaced by equivalents with lower complexity? For example, sorting a collection of n items using bubble sort takes $O(n^2)$ time using a serial algorithm. Sorting using heap sort takes only $O(n \log n)$ time, however, and a parallel implementation of bubble sort requires only $O(n)$ time.
- Can we find a tighter complexity bound available for the model? Often a model's experimental performance is better than the known upper bound, either because the upper bound is loose or because the upper bound depends on circumstances that can be avoided pragmatically. For example, graph-coloring is NP-complete, meaning that one can easily test whether solutions are correct but searching for a correct solution may require exponential time. However, finding a solution is only difficult right at the boundary of colorability; adding a few more colors often makes coloring easy.

ple, the $O(n)$ cost of rebuilding a growing hash table is justified by the expected n accesses of $O(1)$ cost it enables.

[2] Each "clock cycle" may, of course, involve quite a bit of computation on the massively parallel hardware of a brain.

Measure	Approximate Budget	Domain	Approximate Scale	Complexity
Response time to select one of a set of n items	10^2 cycles of computation	choosing a door to enter	a few options	$O(c^n)$
		recognizing a word	vocabulary of thousands	$O(\sqrt{n})$
		noticing a small color patch	millions of visual features	$O(\log n)$
Memory to store a set of n items	10^{14} synapses	working a physics problem	a few models	$O(c^n)$
		relations in a social network	hundreds of friends	$O(n^5)$
		word meanings	vocabulary of thousands	$O(n^3)$
Time to learn about a set of n items	10^9 seconds of experience	arithmetic	a few relations	$O(c^n)$
		word meanings	vocabulary of thousands	$O(n^2)$
		possible body movements	billions of positions	$O(n)$

Figure 1: Any measurement of a cognitive task is associated with one or more limited computational resources. Given estimates of the amount of available resources we wish to spend on this model and the scale of the domain, an approximate upper bound for the reasonable computational complexity of models can be established. The table above shows some examples of such budget/domain relations, filled in with numbers that are highly debatable and a plausible complexity bound. These particulars are not the point: the point is that even changing the budget or domain complexity by an order of magnitude or two will have only a small effect on the plausibility of the complexity bounds.

- Might we instead model a slightly different capability that allows a lower complexity model? For example, consensus between two processes is impossible when messages can be lost. If we change the problem to be consensus with a probability ε of failure, and assume messages are lost independently with probability p, however, then this weaker form of consensus requires only $O(\log p/\log \varepsilon)$ messages.
- Can the model be merged with other models, so that they share the same resources? A costly model of a particular phenomenon is more palatable if it can be mapped onto an interaction between several more general components. For example, the cognitive substrate hypothesis (Cassimatis, 2002) proposes that many cognitive faculties that have been sometimes conceived of as independent modules, such as theory of mind, might be accounted for as combinations of a few more general faculties such as spatial reasoning and sequence learning.

Thus, considering how any particular model potentially contributes to a broad model of intelligence imposes resource constraints that we can use computational complexity to analyze. These constraints are important because even apparently simple models may be computationally intractable. A model with unreasonable complexity demands revision before it can be accepted, just as a phenomenon with weak statistical significance uncovered by human experiment demands new experiments before it can be judged to exist. Nor does the discussion of computational complexity need to be much longer than discussions of statistical significance—in most papers, a paragraph or two plus appropriate references would suffice.

In summary, we argue that a cognitive scientist should no more accept a computational paper with no complexity analysis of its models than an experimental paper with no significance analysis of its data.

Example: word learning

How do questions about computational complexity apply to the two examples we have chosen? Xu & Tenenbaum (Xu & Tenenbaum, 2007), do not suggest a particular implementation for their Bayesian model and explicitly avoid making claims about how the brain performs computations. Nevertheless, complexity analysis may be applied to strengthen this work. If a Bayesian model describes human behavior, then the computations carried out in the human brain must either implement the Bayesian model (albeit likely rather abstractly) or implement something that approximates the Bayesian model. In the absence of a proposal for approximation, however, it is only reasonable that a reader consider the cost of implementing the model as described.

Considering Xu & Tenenbaum's problem of word learning, three basic questions of cost immediately spring to mind:

- How large is the model in memory?
- How costly is it to add new evidence to the model?
- How much does the model grow over a lifetime?

The model proposed in (Xu & Tenenbaum, 2007) is based on tree-structured clustering of a set of *a priori* similarity scores. Although the authors note that "Computing [a similarity] score could be quite difficult..." we will begin with the optimistic assumption that computing the similarity of two objects takes $O(1)$—constant time. For a set of n labelled objects, then, it costs $O(n^2)$ time to find their similarity measures and $O(n^2)$ time to cluster them into a tree-structured taxonomy of $O(n)$ hypotheses (assuming group-average agglomerative clustering). Any new example can potentially change every cluster in the tree, so we assume it is recomputed from scratch every time that a new object is perceived.

Given this tree, the paper describes how the probabilities $p(h)$ can be computed for a cost of $O(1)$ and $p(X|h)$ computed for a cost of $O(|X|)$, where X is the set of examples pertaining to a particular word w. Bayes's rule then allows the likelihood of a hypothesis about any particular word to be computed for $O(|X|n)$, and with n hypotheses from the tree-structured taxonomy, the best can be picked out for $O(|X|n^2)$.

We can now answer the questions above. Since the tree may need to be recomputed from scratch every time, all n labelled objects must be stored. The space for the tree and all annotating information, however, is still only $O(n)$—a very comfortable cost. Again, because everything may be recalculated from scratch, the cost of adjusting the model for a new piece of evidence equals the cost of computing similarity, building the tree, and picking the best hypothesis for a the word category being learned: $O(|X|n^2)$ time. Over a lifetime of n examples, then, the time cost is $O(|X|n^3)$. When n is a few dozen examples, as in the experiments in (Xu & Tenenbaum, 2007), this number is not very big. But if a person encountered around a million labelled examples over their lifetime, then n^3 is 10^{18} and this model begins to look very expensive indeed.

Considering that the authors avoided making any claims about how to perform the computations in their model, the authors are probably cognizant of these issues and have thoughts about how to resolve them. For example, other tree-building algorithms may give much better incremental performance. An explicit discussion of the computational complexity issues, however, improves the work by helping the reader to fairly assess the cognitively plausibility of their model, by clearly indicating open problems implied by complexity issues, and by clearly showing when complexity issues might preclude the reader from extending the model beyond the particular experiments reported.

Example: analogical reasoning

The analogical reasoning architecture described in (K. D. Forbus & Hinrichs, 2006) relates even less directly to the human brain, as it only seeks to cope with a problem that human minds must cope with as well. Nevertheless, relating the complexity of the computational model back to questions of cognitive plausibility can raise important questions for investigation.

In the Companion architecture (which has actually been constructed) a user interacts with a cluster of machines all doing analogical retrieval on a knowledge base that contains models of the domain, tasks, dialog, users, and itself. Brushing aside a number of other details, let us consider two basic activities of the architecture: searching the knowledge base for promising examples and manipulating small numbers of examples with algorithms based on SME analogical mapping (Falkenhainer, Forbus, & Gentner, 1989). We now consider the following three questions:

- What are the complexities of searching the knowledge base and analogical mapping?
- What is the expected scale of the examples and the knowledge base?
- What can we learn from comparing these complexities and domain scales with available biological resources?

Although the paper does not directly address computational complexity, we can mostly answer the first question by following its references. Searching the knowledge base is done by the MAC stage of MAC/FAC retrieval (K. Forbus, Gentner, & Law, 1995), which takes $O(k)$ operations for a knowledge base with k elements. This can be parallelized trivially, resulting in $O(\log k)$ time and $O(k)$ circuit complexity (a form of space complexity). Analogical mapping operations are implemented using a greedy version of SME requiring time linear in the number of kernel mappings, which are bounded by a worst case $O(n^2)$, where n is the number of elements in an example. In practice, however, the complexity appears to often be much lower (K. Forbus, Ferguson, & Gentner, 1994). Mappings are made in the FAC portion of retrieval, which in practice uses only three (K. Forbus, Usher, & Tomai, 2005), and SEQL (Kuehne, Forbus, & Gentner, 2000), where the number of mappings required is unclear, so we optimistically assume that it too is constant.

Next, we look for estimates of the scale of n and k. One of the three domains examined in (K. D. Forbus & Hinrichs, 2006), tactical decision games, has cases with a size of approximately $n = 1000$ propositions (case size in the other two domains is not stated). The expected size of the knowledge base is not specified in this paper, but Forbus and colleagues are architecting it to support knowledge bases with $k = 10^9$ elements (K. D. Forbus, 2009), which gives us a good ballpark estimate of their guess at how large human-scale knowledge might be.

With these established complexities and estimates of scale, we can now consider the relationship of the Companion architecture to analogical reasoning in human minds. We need only assume that human minds are also doing large-scale analogical reasoning, and that the algorithms' costs are more due to the computational nature of analogy than to peculiarities of design—a reasonable conjecture after two decades of pragmatic honing.

Consider the 10^9 scale of the knowledge base. At this scale, logarithmic time is quite reasonable, so long as the constants are not bad. For circuit complexity: 10^9 elements is much smaller than 10^{11} neurons. Most of the operations are multiplications or additions, which might even be supported by synapses—and 10^{15} is much larger still. Consider, however, that biological hardware is often fairly noisy. The amount of noise can be reduced by increasing the amount of hardware per operation, but the cost may rise sharply as allowable error decreases. Considering cognitive plausibility thus motivates a new question about retrieval: how sensitive are the results of MAC/FAC to noise in the MAC stage? If the sensitivity is high enough, then that casts doubt on the plausibility of large-scale knowledge bases and may lead to investigation of less sensitive retrieval methods (which may have other interesting cognitive properties) or of how large-scale analogy might be supported with smaller knowledge bases.

Time is the limiting factor for analogical matching. With n in the range of 10^3, matching might cost on the order of 10^6 serial time steps—many minutes of neuron-speed "clock ticks"—but the common case for the greedy algorithm is expected to be much faster. Empirical surveys can determine whether it is reasonable to expect the common case to be fast enough—$O(\log n)$ would be a nice bound to discover, while $O(\sqrt{n})$ would be pushing it and $O(n)$ probably too much unless the constant is much below one. We might also test whether something like the greedy algorithm is being used by humans: if it is, then an experimenter should be able to produce $O(n^2 + c)$ scaling in human response time by manipulating the similarity of elements (and thereby the number of kernel matches), where c is an "overhead" constant for the portion of the response time not due to an SME-like algorithm.

We thus see that using complexity to relate even a purely computational model, such as the Companion architecture, to cognitive plausibility can yield new insights, constraints on the system, and questions for further investigation.

Benefits of considering complexity

Complexity analysis of cognitive models is also likely to improve the scalability, robustness, composability, and longevity of these models. In computer science, complexity analysis of algorithms helps to mitigate pragmatic difficulties in these areas caused by the nature of the field. We argue that analogous pragmatic difficulties exist in cognitive science, and that complexity analysis of models may have a similarly beneficial effect.

Scaling In computer science, large-scale algorithms can often be tested experimentally only at small scale—consider, for example, peer-to-peer file sharing, internet routing, and database mining. Complexity analysis shows how small scale behavior can be expected to predict large scale behavior.

Pragmatic constraints often similarly limit the experiments that can be performed with computational cognitive science models. For example, in the word-learning experiments in (Xu & Tenenbaum, 2007), subjects learn a category from images of 21 objects. In a lifetime, however, a human encounters orders of magnitude more objects and many different categories. In (K. D. Forbus & Hinrichs, 2006), the analogical reasoner is being applied to three domains, at least one of which contains cases with 10^3 propositions. In daily life, however, a human encounters many more domains to think about, and depending on how a domain is formulated, the number of propositions in a case might be much greater. With knowledge of how the models scale, it is possible to argue that these experimental results imply something more general about word learning or analogical reasoning; without such knowledge, they only provide information about the small scale that has been tested.

Robustness Computers are complex systems where unpredictable interactions between components may interfere with experimental measurements. Complexity analysis dampens the impact of such interference by measuring the scaling of behavior instead of its absolute value.

We hardly need to mention that cognitive experiments are similarly beset with potential interference. Any experiment is subject to interference from the cognitive faculties not under consideration, and the complex interactions between faculties may produce non-random interference that cannot be averaged away. The word-learning experiments in (Xu & Tenenbaum, 2007), for example, use just one word and 45 pictures of objects. How shall we judge that the effects observed are likely to apply to the full human scale of thousands of words and millions or more perceived objects?

The usual approach to ruling out the effects of interference is to brainstorm all of the ways that other systems or processes might interfere ("the confounds") and run an exhaustive series of experiments to rule out each potential confound.

An alternate, complexity-based approach would be to design a set of experiments that vary the amount of evidence provided to the subjects to determine whether their performance scales in the manner predicted by the model. In (Xu & Tenenbaum, 2007), they begin down this path, by training subjects with either one example or three examples, but this says nothing about scaling toward the full problem of many objects and many words. An experiment with variable numbers of words being learned, however, and showing that the judgements scale with the number of words as predicted by the Bayesian model, and that that complexity is reasonable for the full scale problem, would be much more compelling.

Composability of Models An algorithm may be reused in many different contexts, on many different sorts of problems. Complexity analysis shows how behavior in a known context or with a known problem predicts the behavior for novel contexts and problems.

Cognitive models face similar challenges because there are no current whole-mind models that constrain the design of smaller models. This means that the context in which a fragmentary model may operate is unknown, and the scale of the model might vary significantly based on this context. Consider our two example systems, word learning and analogical reasoning. In word learning, for example: how many words must be learned? How many examples are provided, and how much clutter distorts each example? The answers to these questions may vary by several orders of magnitude depending on how a "word" is represented (e.g. what counts as a word sense? must the word be used or simply recognized?) and how the examples are structured (e.g. a constant stream or occasional episodes? filtered or unfiltered? high-level or low-level?). Analogical reasoning is just as uncertain: how many cases are there in a person's memory? How complex are they? How often do they need to be retrieved? How frequently are analogies made?

Without complexity analysis, we do not know what will happen to these models when the scale is changed. Unless a broad model is built on top of a fragmentary model, the fragmentary model probably will not be used within the same context or general constraints that it was originally tested with. Knowledge of how a model scales, however, allows us to predict how it will behave in a new context.

Longevity of Models Computer systems are highly variable in structure and rapidly changing. Complexity analysis allows results about an algorithm to be portable from system to system—even across fundamentally different models of computation (e.g. stack machine vs. register machine vs. stream processor), and predicts how hardware changes will affect the feasibility of an algorithm.

Likewise, experimental evidence is rapidly changing our models of what sort of computations can be supported by the human brain. When a computational model is tied closely to a particular model of neural computation, it becomes embroiled in the controversies of neuroscience and is often quickly rendered obsolete. When a model is disconnected entirely from the notion of biological implementation—as is the case in both (Xu & Tenenbaum, 2007) and (K. D. Forbus & Hinrichs, 2006)—then it must be constantly defended against charges that it may not be realizable in a human brain or may not actually explain what people are doing.

Complexity analysis against an abstract biological hardware model (such as the one presented in (Beal, 2007)), however, shows the realizability of a system without tying it to any particular model of neural computation. Instead, any model of neural computation, even one not yet dreamed of when the cognitive model was created, simply adjusts the constant multipliers of the asymptotic complexity.

Pragmatics of discussing complexity

A presentation of any model in computational cognitive science thus needs to address the following four questions:

- What resource limitations are pertinent to this model?
- What is the order of growth for the model and its operations with respect to these resources?
- What is the scale of the model?
- How does the model compare against the current estimated resource limits of biological intelligence?

This does not mean that publication must wait on precise understanding of complicated models. A complexity upper bound need not be tight, and it is perfectly acceptable to overestimate the resource consumption of a model—even by a vast amount! The bounds can always be tightened later.

The breadth of cognitive science makes it particularly important for computational cognitive science papers to answer all of these questions explicitly and clearly. Many cognitive scientists are not skilled in computational complexity analysis, and thus avoiding a clear statement of a model's computational complexity and its implications is likely to accidentally mislead one's colleagues by "sweeping under the rug" the computational difficulties of the model. Omitting complexity analysis stifles research by making it seem as though open problems have already been solved.

Contributions

We have argued that computational cognitive scientists should explicitly analyze the computational complexity of their models. To demonstrate this, we have shown how analysis of computational complexity can be applied to strengthen Tenenbaum's work on Bayesian learning and Forbus' work on analogical reasoning.

We expect that treating computational complexity as a primary issue, with the same importance as significance analysis in experimental work, will greatly enhance the ability of cognitive scientists to communicate and to compare work across paradigms, and we strongly encourage the community to adopt it as a standard of evaluation for computational research.

Acknowledgements

The authors thank Ken Forbus for his help and patience in sorting out details about analogical reasoning, and Josh Tenenbaum for his feedback. J.M.R thanks the Fannie and John Hertz foundation and both authors thank the National Science Foundation for their financial support.

References

Beal, J. (2007). Developmental cost for models of intelligence. In *Aaai 2007 workshop on evaluating architectures for intelligence.* AAAI.

Cassimatis, N. (2002). *Polyscheme: A cognitive architecture for integrating multiple representation and inference schemes*. Unpublished doctoral dissertation, MIT.

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

Forbus, K., Ferguson, R., & Gentner, D. (1994). Incremental structure-mapping. In *Sixteenth annual conference of the cognitive science society.*

Forbus, K., Gentner, D., & Law, K. (1995). Mac/fac: A model of similarity-based retrieval. *Cognitive Science*, *19*(2), 141–205.

Forbus, K., Usher, J., & Tomai, E. (2005). Analogical learning of visual/conceptual relationships in sketches. In *Aaai-05.*

Forbus, K. D. (2009). personal communication.

Forbus, K. D., & Hinrichs, T. R. (2006). Companion cognitive systems: A step toward human-level ai. *AI Magazine*, *27*(2).

Kuehne, S., Forbus, K., & Gentner, D. (2000). Seql: Category learning as incremental abstraction using structure mapping. In *Cognitive science conference.* Cognitive Science Society.

Shepherd, G. (2004). *The synaptic organization of the brain*. Oxford University Press.

Williams, R. W., & Herrup, K. (1988). The control of neuron number. *Annual Review of Neuroscience*, *11*, 423–453.

Xu, F., & Tenenbaum, J. (2007). Word learning as bayesian inference. *Psychological Review*, *114*(2).