2013 Annual Conference on Advances in Cognitive Systems: Workshop on Metacognition in Situated Agents

Towards A Unified Framework for Learning and Processing Perceptual, Relational, and Meta Knowledge

Marc Pickett MARCPICKETT1@GMAIL.COM
NRC Postdoctoral Fellow, Naval Research Laboratory, Washington, DC

Abstract

We present a framework for representing, learning, and processing meta-knowledge. Our framework leverages a system initially developed for perceptual learning and processing to process both relational structures in general and feature hierarchies in particular. We describe preliminary results demonstrating how our system can be used to learn *meta-ontologies*, or feature hierarchies of feature hierarchies.

1. Introduction

A major difference between the minds of humans and other mammals is humans' ability to perform metacognition. Though there is debate about the level of non-human mammals' metacognitive capabilities, it is generally agreed that humans' abilities vastly surpass those of other animals in this area (Carruthers, 2008; Smith, 2009). However, in terms of gross neuroanatomy, human brains seem to have no special structures or mechanisms that are absent in the brains of simpler mammals, such as rabbits, that have little cognitive capacity beyond perception and action (Roth & Dicke, 2005). The chief difference between human brains and those of other mammals is that humans have a vastly expanded neocortex (Rilling, 2006). Furthermore, there is evidence that the neocortex in newborns is both uniform and plastic, with differentiation arising through learning from experiences (Sur & Rubenstein, 2005; Mountcastle, 1978). That is, the a newborn's neocortex is the same basic mechanism repeated many times, and it is this mechanism, which we call the *cortical substrate*, that accounts for the bulk of human learning and reasoning.

From this evidence, we adopt the hypothesis that it is possible to build an intelligent "newborn" agent using only a handful of basic mechanisms. If this hypothesis is true, such an elegant design is attractive for researchers studying cognitive systems because creating an agent with human-level intelligence would entail an implementation of only a handful of mechanisms (and allowing the system to develop its representations through learning) rather than specialized mechanisms for each of the myriad aspects of human intelligence.

Although our work is not constrained by biological plausibility, algorithms loosely based on the neocortex have emerged that have desirable properties for intelligent agents. For example, cortically-inspired models have achieved state-of-the-art performance for computer vision (Le et al., 2012) and some classification tasks (Chandrashekar & Granger, 2011) by learning feature hierarchies.

If an expanded neocortex accounts for the bulk of the cognitive differences between humans and other mammals, then an open question is how an expanded neocortex might account for these differences. That is, an account is missing of how a cortical substrate can be leveraged to account for higher level cognition, such as symbolic reasoning, analogical inference, and metacognition (Granger, 2011).

In previous work, we showed how a cortically-inspired algorithm could account for analogical inference (Pickett & Aha, 2013b). In this paper, we present preliminary results in our attempt to extend our earlier framework to support meta-knowledge, with the belief that this will be useful for metacognitive processing. In particular, we focus on the the question of how knowledge *about* feature hierarchies can be encoded as a feature hierarchy.

First, we provide background on *Ontol*, a model loosely based on the cortical substrate, and how it can be used to process both perceptual and relational data (Section 2). We then show how Ontol can be used to build a hierarchy of feature hierarchies (Section 3). Finally, in Section 4 we discuss shortcomings of our framework and how it might be applied to metacognition, and then conclude (Section 5).

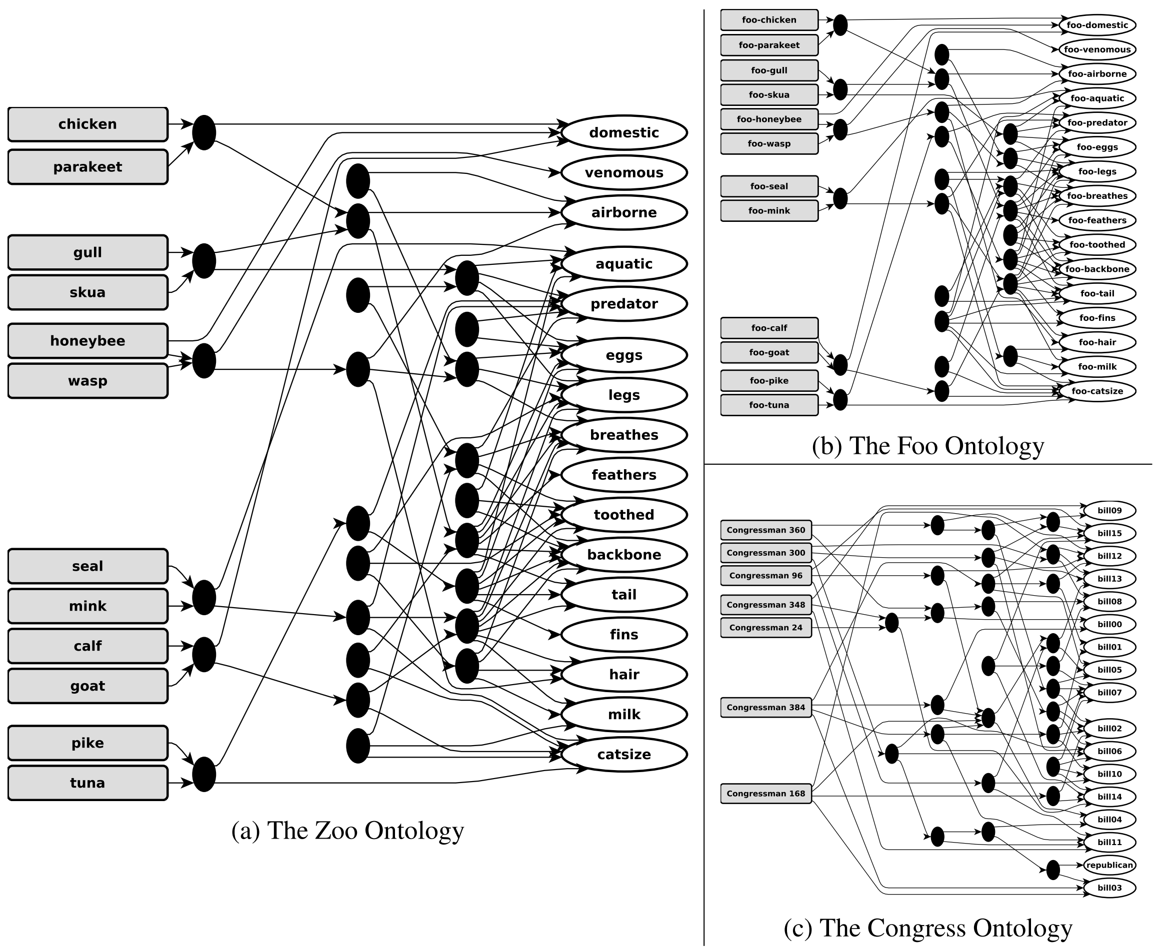
2. Background on Learning Feature Hierarchies

Our current model for the cortical substrate, *Ontol* (Pickett, 2011), is a pair of algorithms, both of which are given "sensor" inputs (fixed-length, real-valued non-negative vectors). The first algorithm, chunk, constructs a feature hierarchy, or *ontology*, that concisely encodes the inputs. For example, given a set of vectors representing visual windows from natural images, Ontol produces a feature hierarchy loosely modeled on that seen in the visual cortex. The second algorithm, parse, takes as input an ontology (produced by the first algorithm) and a new vector, and *parses* the vector. That is, it produces as output the new vector encoded in the higher-level features of the ontology. In addition to "bottom-up" parsing, the second algorithm also makes "top-down" predictions about any unspecified values in the vector.

Ontol is ignorant of the modality of its input. That is, Ontol is given no information about what sensory organ is producing its inputs. Because of this ignorance, we are able to leverage Ontol to find patterns in abstract "sensory" inputs that are actually encodings of relational structures.

Figure 1a shows the ontology constructed by chunk when applied to an animal dataset (from Blake & Merz, 1998), where the "sensory percepts" are features for each animal. For simplicity, in this example and the remainder of the paper, we will consider a simplified version of Ontol that takes as inputs *feature bags*, which are mathematically equivalent to sparse vectors with positive integer values.

In earlier work we demonstrated how relational structures, such as the "Sour Grapes" story (from Thagard et al., 1990) shown in Figure 2, can be represented as feature bags such that they could be given to Ontol as input. When given a collection of stories represented this way, Ontol learned a hierarchy of plot devices (shown in Figure 3) that could be used to efficiently retrieve analogs for new stories (Pickett & Aha, 2013a) and perform analogical inference (Pickett & Aha, 2013b).



(a) The Zoo Ontology

(b) The Foo Ontology

(c) The Congress Ontology

Figure 1: Structurally Similar Ontologies. Instances are shown on the left, and base-level features are on the right. Black nodes in the middle correspond to higher-level features. For example, in Figure 1a, the `chicken` is described by the black node roughly corresponding to "domestic fowl", which, in turn, is described as both `domestic` and by the node that roughly corresponds to "bird".

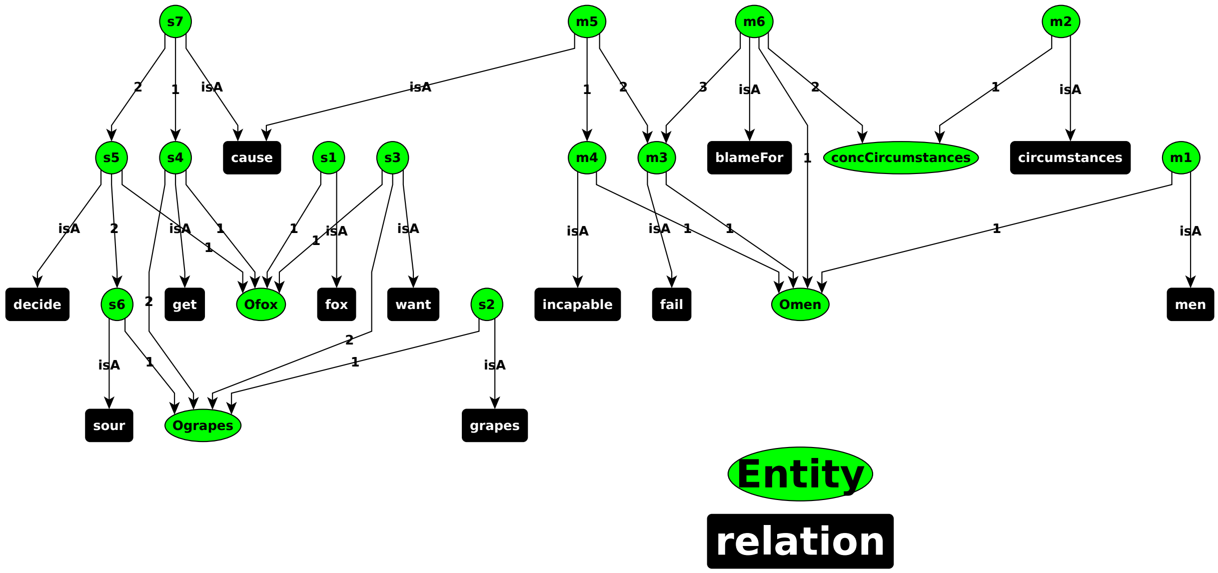


Figure 2: The “Sour Grapes” Relational Structure. This fable was converted into predicate form by Thagard et al. 1990, and displayed here as a graph, where relations are shown as black rectangles with numbered edges and entities are shown as green ellipses. For example, the structure beginning with the node at top center-left marked as `m6` represents the statement `(blameFor Omen concCircumstances m3)`, which can further be expanded to `(blameFor Omen concCircumstances (fail Omen))`.

3. Learning Hierarchies of Feature Hierarchies

The key insight for this paper is that ontologies, such as those shown in Figures 1a and 3, themselves are relational structures. Therefore, we can encode these structures using a method similar to that for encoding stories, and use Ontol to learn an ontology that describes a group of ontologies. Although many methods have been proposed for representing meta-knowledge (for a survey see Cox, 2005), to our knowledge, the work presented here is the first representation that uses a feature hierarchy to represent both perceptual knowledge, higher-order knowledge, and knowledge about knowledge.

We now describe a method for transforming ontologies into feature bags so that these can be given as input to Ontol. The first intuition behind this transform is that a feature hierarchy essentially encodes relations among the nodes in it. The second intuition is that an encoding of relations among n entities can be approximated by describing the relations among the entities in each of many overlapping subsets of n. For example, if $hair$ is highly correlated with event $milk$ and $milk$ is highly correlated with $warmblooded$, then it is often the case that $hair$ and $milk$ together predict $warmblooded$. For the current implementation, we use tallies over truth values to represent relations. For example, the 3-way relation among $hair$, $milk$, and $warmblooded$ would be represented by a truth table with a tally for each of the 2^3 truth assignment possibilities.

The size of the truth tally is exponential in the number of variables. Therefore, we break each large relational structure into multiple overlapping *subsets* of nodes. Our algorithm exploits a principle akin to one used by the HMax model of the visual cortex (Riesenhuber & Poggio, 1999): as the number of subsets for a relational structure increases, the probability decreases that another structure has the same subsets without being isomorphic to the first.

The process for learning a meta-ontology from a set of ontologies, called metaOntol, is described in Figure 1. This algorithm extracts t subsets from each ontology, transforms them into canonicalized truth tables. Treating each truth table as a feature bag and metaOntol chunks these feature bags to create an ontology of truth tallies called $truthTallyOntology$. metaOntol then re-encodes the truth tallies by parsing them using this ontology, and re-encodes the original ontologies (from which the subsets came) as a feature bag of the parsed windows. Finally, metaOntol runs another pass of Ontol's chunking algorithm on the re-encoded structures to generate the meta-ontology. Note that since Ontol is ignorant of the modality of its input, it processes this meta-knowledge exactly the same way it processes any other type of knowledge.

As a preliminary proof of concept, we created a simple ontology of ontologies by first learning 6 object-level ontologies, transforming these to feature bags, then learning a meta-ontology from the transformed ontologies. The 6 object-level ontologies included the Zoo ontology shown in Figure 1a, a "Congress" ontology partially shown in Figure 1c, which Ontol learned from a dataset of 435 congressmen's votes on 16 bills in 1984. We also learned two additional ontologies from copies of each of these datasets, with the exception that the instances and features have been renamed. For example, from the "zoo" dataset we created the "foo" dataset in which a `foo-chicken` is `foo-domestic`. As shown in Figure 1b, the resulting ontologies are structurally similar (though not completely isomorphic due to randomization effects) to the original ontologies, but superficially dissimilar because they share no common nodes. (We assume that for our system, the atomic symbols `foo-domestic` and `domestic` look no more alike than any other pair of symbols.)

Given these 6 ontologies (and somewhat arbitrarily choosing the size of truth tally $m = 4$, and the number of truth tallies per ontology $t = 100$), metaOntol transformed each into a bag of features using the method described above. Using this representation Ontol learned the ontology shown in Figure 4. This meta-ontology successfully groups together the structurally similar but superficially different "zoo", "foo", and "voo" ontologies, as well as the variations of the "congress" ontologies.

4. Discussion

The framework we have presented, using cortically-inspired models to represent knowledge about cortical models, is still new and currently has much room for growth. In particular, further work is needed to investigate *how* meta-ontologies might be used by an intelligent agent to accomplish its goals. We also discuss how the current framework might be improved.

It is interesting to consider how the meta-ontology in Figure 4 might be used. Since the meta-ontology captures structural, rather than surface overlap, the meta-ontology might be used for analogical knowledge transfer between object-level ontologies. For example, a similar meta-ontology might be used to find invariances in computer vision and other modalities by finding areas that are superficially different yet behaviorally similar. (E.g., by noticing that the feature hierarchy describing the top left of the visual field is roughly isomorphic to the hierarchy describing the bottom right, a system could discover translation invariance.)

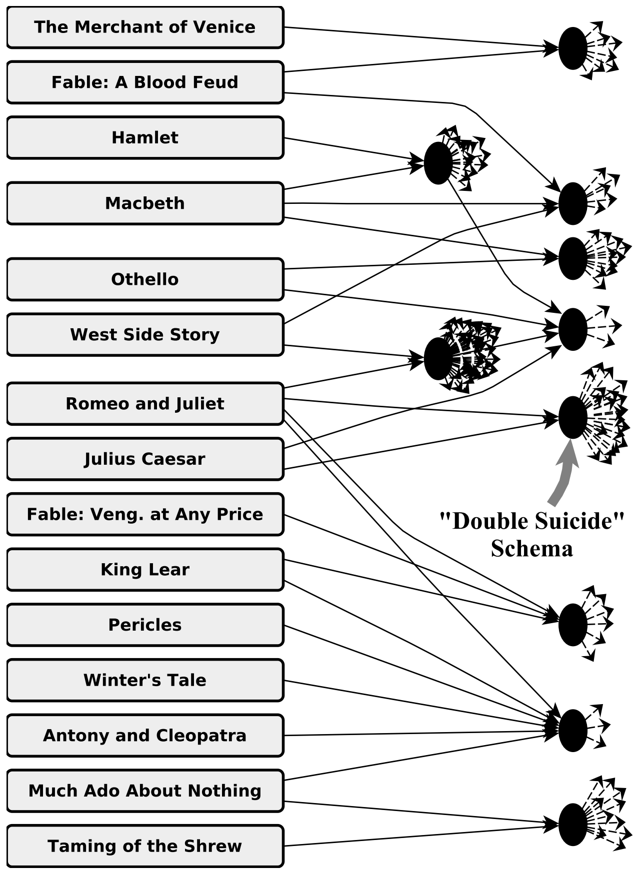


Figure 3: Part of the ontology learned from a story dataset. As in the Zoo Ontology in Figure 1a, black ovals represent higher level concepts. While in the Zoo Ontology, higher level concepts correspond to shared surface features, in this figure, high level concepts correspond to shared structural features (plot devices). For example, the denoted oval on the right represents a *Double Suicide* schema, which happens in both *Romeo and Juliet* and in *Julius Caesar*.

Table 1: Algorithm for Learning a Meta-ontology

```
// Input is Ω, a set of ontologies and I, a set of instances
// m is the size of truth tally, and t is the number of truth tallies per ontology
define metaOntol(Ω, I, m, t):
   // Create an ontology C of canonicalized truth tallies.
   let allTallies be an empty list.
   // Get the truth tallies.
   foreach ω ∈ Ω:
      let tallies_ω be an empty list.
      // Get t truth tallies.
      repeat t times:
         let truthCounts be a hash tally initialized to all zeroes.
         let smallont = randomly chosen subset of ω of size m
         for i ∈ I:
            let on_i = the set of nodes in smallont activated when using ω to parse i
            truthCounts[on_i] + + // Make tally for truth entries
         add canonicalize(truthCounts, smallont) to allTallies and to tallies_ω
   // Chunk the truth tallies
   let truthTallyOntology =chunk(allTallies)

   // Represent each ω as a count of parsed tallies
   let ΩasSet be an initially empty list.
   foreach ω ∈ Ω:
      let ωasTallyOfTallies be an initially empty set.
      foreach truthCounts ∈ tallies_ω:
         parse truthCounts using truthTallyOntology and add the parse to ωasTallyOfTallies
      add ωasTallyOfTallies to ΩasSet

   // Chunk the set of ontologies represented as feature bags.
   return chunk(ΩasSet)

// Routine to canonicalize a truth tally.
// This tries all m! orderings of smallont
define canonicalize(truthCounts, smallont):
   // Find the ordering that yields the truth-tally that comes first "alphabetically".
   let bestCounts = truthCounts
   foreach ordering in all m! orderings of smallont:
         make a new truth tally truthCountsNew substituting ordering for smallont
         if comesFirst(truthCountsNew, bestCounts)
   return bestCounts

// Comparing two isomorphic truth tallies.
define comesFirst(a, b):
   let keys be the sorted keys from the union of a's and b's keys
   foreach key ∈ keys:
      if a[key] ≠ b[key]: return a[key] ≠ b[key]
   return false
```

Motor controllers, such as traces of arm movements, can be encoded in the cortex of the cerebellum (Albus, 1971). Conceivably, we can create an ontology of such motor controllers and leverage

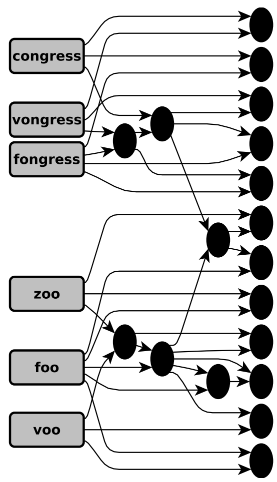


Figure 4: An ontology of ontologies

this ontology to quickly learn new motor skills. If traces of thought can be described as predicate logic, then we can leverage our framework to build an "ontology of thought processes" by transforming predicate logic into a relational structure and then into feature bags. Ontol can then find patterns in these feature bags.

This latter approach may be a promising direction for research in metacognition, as it offers a potential way for a system to analyze its own cognitive processes. This would entail first developing a way to represent cognitive processes themselves as cortical substrate (perhaps in a manner analogous to how motor processes can be represented as sections of cerebellar cortex), then using the current framework to find patterns in the cognitive processes, and finally using these patterns to improve the cognitive processes.

There is room for improvement in both our cortical model and the method for transforming relational structures in general, and ontologies in particular, into feature bags. In its current incarnation, Ontol builds a hierarchy of conjunctions. Significant leverage can be had by including nodes that represent disjunctions (Riesenhuber & Poggio, 1999) and sequences (George & Hawkins, 2009). Including disjunctions will not significantly affect our current transformation algorithm, since truth tallies can be computed using essentially the same method. However, the addition of sequences will require different ways of expressing relations among nodes, such as graphlet kernels (Shervashidze et al., 2009). metaOntol currently assumes that the separate ontologies are both segmented and within a manageable size. In reality, ontologies such as semantic networks tend to form contiguous networks (Steyvers & Tenenbaum, 2005). To address both problems, a future version of our transformation will grab a hierarchy of overlapping bounded-size contiguous subgraphs from the ontology in a manner analogous to overlapping receptive fields in computer vision.

5. Conclusion

In this paper we have shown preliminary results demonstrating how Ontol, a system capable of learning feature hierarchies, can learn a meta-ontology, or hierarchy of feature hierarchies. This work is still in its early stages, and future work includes detailing *how* meta-ontologies can be used for knowledge transfer, analogical reasoning, invariance discovery, and meta-cognition.

References

Albus, J. S. (1971). A theory of cerebellar function. *Mathematical Biosciences*, *10*, 25–61.

Blake, C., & Merz, C. (1998). UCI Repository of Machine Learning Databases.

Carruthers, P. (2008). Meta-cognition in animals: A skeptical look. *Mind & Language*, *23*, 58–89.

Chandrashekar, A., & Granger, R. (2011). Derivation of a novel efficient supervised learning algorithm from cortical-subcortical loops. *Frontiers in computational neuroscience*, *5*.

Cox, M. T. (2005). Metacognition in computation: A selected research review. *Artificial Intelligence*, *169*, 104–141.

George, D., & Hawkins, J. (2009). Towards a Mathematical Theory of Cortical Micro-circuits. *PLoS Comput Biol*, *5*.

Granger, R. (2011). How brains are built: Principles of computational neuroscience. *Cerebrum; The Dana Foundation*.

Le, Q., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G., Dean, J., & Ng, A. (2012). Building High-Level Features using Large Scale Unsupervised Learning. *International Conference in Machine Learning*.

Mountcastle, V. (1978). An Organizing Principle for Cerebral Function: The Unit Model and the Distributed System. *The Mindful Brain*, 7–50.

Pickett, M. (2011). *Towards Relational Concept Formation From Undifferentiated Sensor Data*. Doctoral dissertation, University of Maryland Baltimore County.

Pickett, M., & Aha, D. (2013a). Spontaneous Analogy by Piggybacking on a Perceptual System. *Proceedings of the 35th Annual Conference of the Cognitive Science Society* (pp. 3229–3234). Austin, TX.

Pickett, M., & Aha, D. W. (2013b). Using cortically-inspired algorithms for analogical learning and reasoning. *Biologically Inspired Cognitive Architectures*, *6*, 76–86.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical Models of Object Recognition in Cortex. *Nature Neuroscience*, *2*, 1019–1025.

Rilling, J. K. (2006). Human and Nonhuman Primate Brains: Are they Allometrically Scaled Versions of the Same Design? *Evolutionary Anthropology: Issues, News, and Reviews*, *15*, 65–77.

Roth, G., & Dicke, U. (2005). Evolution of the Brain and Intelligence. *Trends in Cognitive Sciences*, *9*, 250–257.

Shervashidze, N., Vishwanathan, S., Petri, T., Mehlhorn, K., & Borgwardt, K. (2009). Efficient Graphlet Kernels for Large Graph Comparison. *Int. Conf. on AI & Stats.*.

Smith, J. D. (2009). The study of animal metacognition. *Trends in cognitive sciences*, *13*, 389–396.

Steyvers, M., & Tenenbaum, J. B. (2005). The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. *Cognitive science*, *29*, 41–78.

Sur, M., & Rubenstein, J. L. (2005). Patterning and plasticity of the cerebral cortex. *Science Signaling*, *310*, 805.

Thagard, P., Holyoak, K., Nelson, G., & Gochfeld, D. (1990). Analog Retrieval by Constraint Satisfaction. *Artificial Intelligence*, *46*, 259–310.