Hierarchical Relation Networks: Exploiting Categorical Structure in Neural Relational Reasoning

Ruomu Zou[1] and Constantine Dovrolis[2]

[1]*Phillips Exeter Academy, Exeter NH, U.S.A.*

[2]*School of Computer Science, Georgia Tech, Atlanta GA, U.S.A.*

Keywords: Relational Reasoning, Categorization, Hierarchy Learning.

Abstract: Organizing objects in the world into conceptual hierarchies is a key part of human cognition and general intelligence. It allows us to efficiently reason about complex and novel situations relying on relationships between object categories and hierarchies. Learning relationships among sets of objects from data is known as relation learning. Recent developments in this area using neural networks have enabled answering complex questions posed on sets of objects. Previous approaches operate directly on objects – instead of categories of objects. In this position paper, we make the case for reasoning at the level of object categories, and we propose the *Hierarchical Relation Network (HRN)* framework. HRNs first infer a category for each object to drastically decrease the number of relationships that need to be learned. An HRN consists of a number of distinct modules, each of which can be initialized as a simple arithmetic operation, a supervised or unsupervised model, or as part of a fully differentiable network. This approach demonstrates that categories in relational reasoning can allow for major reductions in training time, increased data efficiency, and better interpretability of the network's reasoning process.

1 INTRODUCTION

A key component of human intelligence is the ability of associating entities (such as your best friend's younger brother or the chair in your grandmother's living room with the embroidered cushion) with categories such as people or furniture. (Alexander et al., 2016; Krawczyk, 2012). These categories are also often referred to as "concepts" which we form to better abstract our world (Pothos and Wills, 2011; Rosch et al., 1976; Markman and Wisniewski, 1997; Yeung and Leung, 2006). Forming categories allows us to carry out efficient and robust relational reasoning.

Relational reasoning considers problems that require assessing not only individual objects in a dataset, but how groups of such objects relate. For example, looking at an office desk like the one in Figure 1 on the following page, one might ask the non-relational question "is the lamp on?" This question only looks at one object: the lamp. A relational query, on the other hand, would consider more objects – for example: "does the lamp shine light on objects which are beside each other?" To answer this query, one needs to compute the relationships between the lamp, the desk, and the chair. To tackle similar challenges, several models have been proposed (Shanahan et al., 2020; Kazemi and Poole, 2018; Sourek et al., 2015; França et al., 2014), the most versatile of which is the Relation Network (RN) architecture, summarized in the next section (Santoro et al., 2017).

Using categories when reasoning about relationships is particularly useful because the number of categories is often much smaller than the number of distinct objects. For example, instead of learning a relation (e.g., "sweeter than") between every type of fruit and every type of vegetable, we can simply learn that "fruits are sweeter than vegetables" – and apply that relation between these two categories of objects to all their corresponding instances. This would allow a learner to deduce that "bananas are sweeter than eggplants" even if the learner was not trained on the (bananas, eggplants) pair.

To test the approach of utilizing categories for neural relational reasoning, we propose the Hierarchical Relation Network (HRN) framework as a proof-of-concept. We demonstrate HRN's plausibility , massively reducing training time without sacrificing accuracy, on a real-world dataset of complex relational queries.

Zou, R. and Dovrolis, C.
Hierarchical Relation Networks: Exploiting Categorical Structure in Neural Relational Reasoning.
DOI: 10.5220/0010690500003063
In *Proceedings of the 13th International Joint Conference on Computational Intelligence (IJCCI 2021)*, pages 359-365
ISBN: 978-989-758-534-0; ISSN: 2184-3236

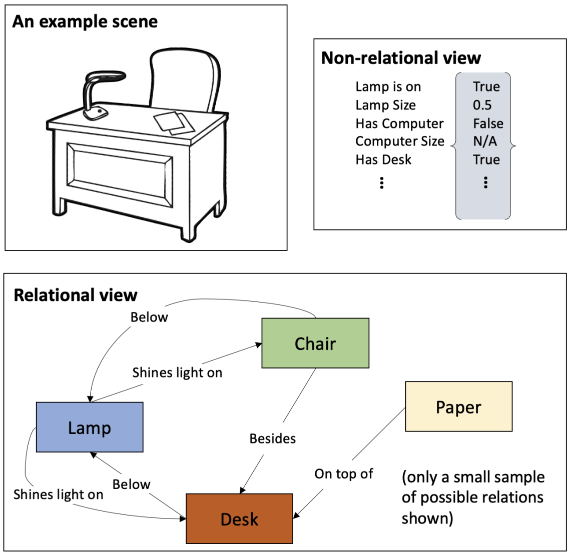

Figure 1: An example of a relational view of a scene vs. a non-relational view.

2 PRIOR WORK

Relation Networks: The Relation Network (RN) architecture was proposed as a general way of operating on inputs represented as sets of objects (Santoro et al., 2017). It takes as input a set of objects and possibly some additional information, such as questions posed on the set of objects. To get the final output (an answer for the posed question), the objects are placed into pairs and each pair is processed separately. This is equivalent to considering the binary relations between every possible object pair. If there are n objects, then the network would have to evaluate n^2 relations. This can quickly get intractable, especially when working with complex domains.

Hierarchy Learning and Semantic Computing: A variety of techniques in semantic computing learns and utilizes hierarchies/categories among objects (Wang, 2010). The algorithm *Concept Semantic Hierarchy Learning (CSH Learning)* was proposed to construct ontologies of concepts for semantic comprehension by machine learning systems (Valipour and Wang, 2017). A similar algorithm, proposed by Anoop et al., constructs concept hierarchies in an unsupervised manner from natural language text corpuses (Anoop et al., 2016). Although these techniques don't focus on relational question answering, they nevertheless demonstrate the value of learning and using hierarchies as opposed to flat lists of concepts.

Categorization in Cognition: Humans naturally organize perceived objects into categories. Past research has identified three main theories for human categorization: classical, prototype, and examplar (Pothos and Wills, 2011). We mainly utilize ideas from the prototype view of classes, which considers each class to be represented by a prototype vector (Yeung and Leung, 2006). Additionally, it has also been shown that instead of overly broad (e.g. all living things) or specific (blue-cream calico British Shorthair) classes, humans often focus on basic categories (like cat) that are neither too general nor too restrictive (Rosch et al., 1976; Markman and Wisniewski, 1997). This supports our usage of one level of object hierarchy (each class belongs to a category) as an efficient simplification to full ontologies.

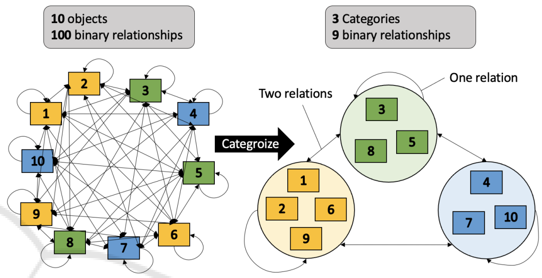

Figure 2: A visual demonstration of using categories to more effectively conduct relational reasoning.

3 HIERARCHICAL RELATION NETWORKS (HRN)

In this section we describe a general framework, termed the Hierarchical Relation Network (HRN), that can harness hierarchical structure among input objects to conduct efficient relational reasoning. It first classifies input objects to their categories, and it then applies the previously proposed Relation Network approach (Santoro et al., 2017) on those categories. Three specific instantiations of this HRN framework are presented, ranging from separate models to a single fully-differentiable neural network.

The process of hierarchical relation reasoning can be divided into three steps which are carried out by three distinct components in the reasoning pipeline. The first component, called a *Categorizer*, assigns each input object into a category. The second component, called a *Conceptualizer*, uses the objects and their predicted categories to generate a dense representation for each category. The third and final component is a network architecture that handles the relational computation, such as a Relation Network (Santoro et al., 2017).

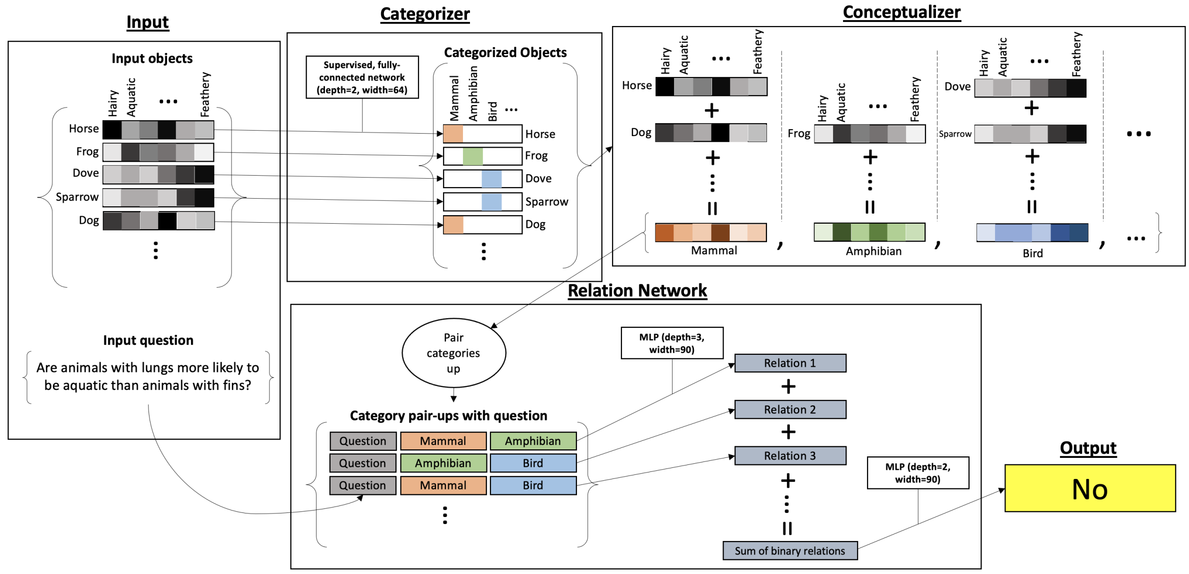

Figure 3: A graphical illustration of an SCHRN with sample input from the Animals dataset below. The raw inputs are first categorized, and then dense representations of each class/concept are built by summing the representations of all objects classified under it. Finally the dense representation for each class is passed to an existing framework for neural relational reasoning (the relational network).

3.1 Supervised Categorizer Hierarchical Relation Network (SCHRN)

The first and simplest implementation of the Hierarchical Relation Network framework takes the form of two separate network models, one for assigning objects to their classes and the other for conducting reasoning (see Figure 3). This implementation is less powerful than the ones presented below, but serves as an important first step. The strict assumptions of SCHRN are relaxed in the subsequent implementations.

To see how SCHRN works, imagine a toddler who is just beginning to learn about the world. The parent may show a number of apples and pears, and say that apples is one type of fruit and pears is another. Through such supervised input, the toddler learns the concepts (object categories) of apples and pears and use these concepts to reason.

The first part of our framework, the *categorizer model*, acts similar to the toddler in the previous example, and it learns object categories through corresponding labeled examples. When the *categorizer model* has been learned, we pass input objects (such as horse, frog etc) to transform them into one-hot representations in the space of categories (mammal, amphibian, etc).

The *conceptualizer* consists of a simple dot product between the previous one-hot representation and the original representation of an input object. This operation results in a category representation that is the sum of the attributes of all member objects of that category. This is similar to the cobweb system for conceptual clustering proposed (Fisher, 1987).

Finally, the previous dense category representations are passed to a network for relational reasoning – in our implementation we use the Relation Network (Santoro et al., 2017) but other networks could also be used.

The main advantage of splitting categorization and reasoning into two networks, apart from more closely resembling human learning, is the additional flexibility introduced. For example, if we only have enough information about an object to determine its category, but no information on its relevant attributes, the SCHRN is effectively unhindered while the subsequent HRN models might suffer depending on their generalization ability.

However, the two-network structure of SCHRN also has drawbacks: it assumes prior knowledge about the categories of some objects used as training data for the categorizer network. It also assumes that the available categories will be relevant to the particular reasoning task (e.g., classifying fruits into ”large fruits” versus ”small fruits” does not help in determining their sweetness). SCHRNs perform well when the previous assumptions are met.

The model UCHRN of the following subsection relaxes the first aforementioned assumption, while

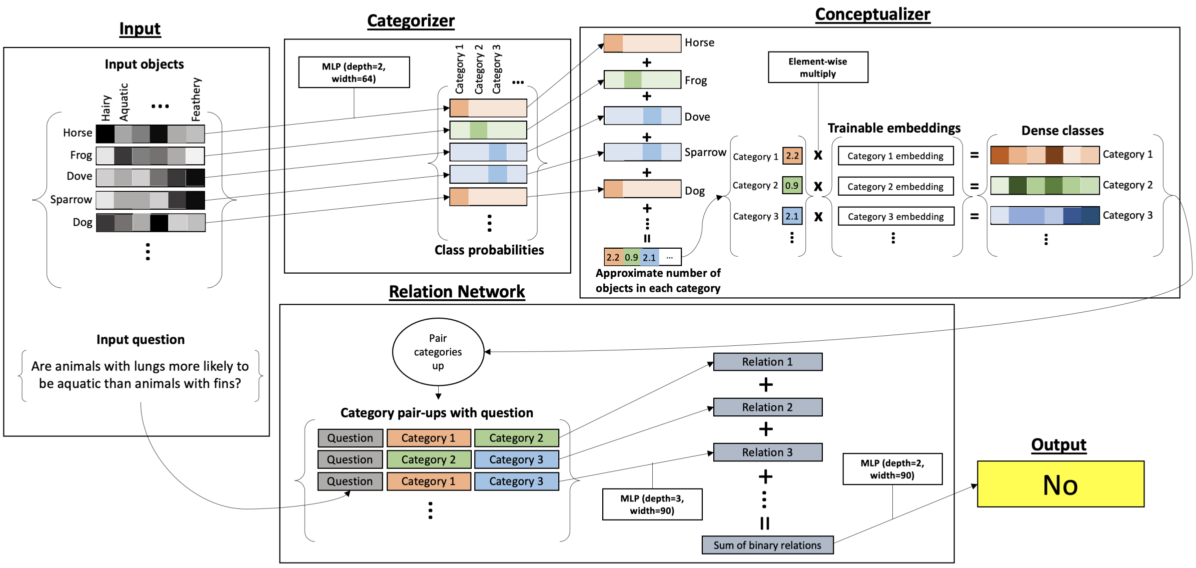

Figure 4: An FDHRN with sample input from the Animals dataset below. The entire framework is end-to-end differentiable. The inputs are first passed through fully connected layers with a softmax activation at the end to get the predicted probability distribution over categories. The probabilities are then summed, yielding the approximate number of objects in each category. They are then multiplied with an embedding matrix containing trainable class embeddings to get the dense representations that are passed into the relation network module.

the model FDHRN of subsection 3.3 relaxes both assumptions. This increase in generality and power comes at the cost of higher variance over the training data however.

3.2 Unsupervised Categorizer Hierarchical Relation Network (UCHRN)

This architecture is effectively identical to SCHRN with one important difference: instead of using a supervised model for categorization, an unsupervised method based on object similarity is employed. This approach is more aligned with the prototype view of cognitive categorization (Yeung and Leung, 2006). Here we use k-means clustering on the object representation to map the raw objects onto one-hot vectors in the space of categories.

Continuing the toddler and fruits analogy, the UCHRN architecture corresponds to the toddler looking at a number of different fruits (apples, pears, melons, etc.), and categorizing them without supervision based on view, taste, smell etc.

An important advantage of UCHRN is that it allows us to vary the number of categories used by the HRN by varying the number of clusters. For convenience, we denote the number of categories by m. This gives us a wide range of intermediate models between the extremes of operating on uncategorized objects ($m = n$, where n is the number of objects) and treating all objects as one category ($m = 1$). In the Results section, we see that the accuracy of UCHRN increases up to a certain point as m increases, before starting to drop.

The UCHRN architecture relaxes the assumption of requiring labeled data for the categorizer. However, it still assumes that the categories found by clustering are good in answering the questions in the test dataset.

3.3 Fully Differentiable Hierarchical Relation Network (FDHRN)

Our third and most general architecture of the HRN framework takes the form of an end-to-end differentiable neural network. Its goal is to form artificial categories/concepts (in the form of embeddings) that are most relevant to answering the questions posed in the training data.

The FDHRN architecture is shown in Figure 4 above. It randomly initializes m categories, the embeddings of which will be trained with the rest of the network. Each input object is first passed through a number of fully connected layers and a softmax function to estimate the likelihood that it belongs to each derived class. The dense representation for each class, which is then passed into the relational reasoning module, is scaled up by summing the "soft" (probabilistic) values of the number of objects it contains.

Table 1: The training time and test accuracy of a baseline RN and three HRN architectures at 30 and 200 epochs.

Architecture	Training time (30 epochs)	Test accuracy (30 epochs)	Training time (200 epochs)	Test accuracy 200 (epochs)
RN (baseline)	2,080s	90.3%	13,349s	92.7%
SCHRN	81s	88.9%	549s	93.3%
UCHRN	67s	90.4%	449s	94.1%
FDHRN	90s	87.8%	545s	89.6%

The intuition behind this approach is that instead of pre-assigning each object to a definite category, we treat each object as belonging, say 95% to category-one, 3% to category-five, and so on. This transition from using hard category assignments, similar to Aristotelic logic, to using soft assignments, similar to fuzzy logic, both allows for differentiability and increases the fitting power of the network. For small m, when many objects would fall ”on the fence” between two classes, the increase in fitting power would result in greater flexibility and improved performance at the cost of training more parameters.

An important feature of the FDHRN architecture is the use of trainable embeddings for the object categories. Imagine a zoo at which we only ask questions about features that have little to do with animal taxonomy, such as color or size (”do brown animals tend to be bigger than black animals?”). If we were to groups objects by more domain-based categories, such as ”amphibian”, ”mammal” etc., there would be a mismatch between the relation questions we ask and the given categories, which would decrease the learning efficiency. Trainable embeddings allow the FDHRN to derive categories that are more specific to the reasoning task at hand.

Although advantageous, inferring categories from data can lead to increased model complexity and make the network more prone to overfitting.

4 RESULTS

4.1 The Zoo Animals Dataset

We used a dataset containing information about 101 animals with various attributes for each animal (Learning, 2020).

We tested all three aforementioned HRN architectures in tasks of relational queries on subsets of the 101 animals. An example query might be: ”Among these animals, are those with lungs more likely to be aquatic than those with fins?” The possible answers (balanced to occur with about the same frequency) are ”yes,” ”no,” and ”about equally likely.”

Each data point consists of a single task: a subset of animals and a query to which the network generates an answer. During training, the correct answer is also provided as a label. The models are scored based on the percentage of correctly performed tasks, with a random guess baseline achieving around 33% accuracy.

4.2 Testing Specifications

We use a version of the Animals dataset that contains about $19,000$ questions on different subsets of animals. Half the questions are used for training while the other half is reserved for testing. Each subset of animals consists of 25 animals. The choice of 25 is arbitrary, and purely made to ensure that the baseline RN completes training in reasonable time. Even so, the RN model takes several hours to train while the three HRN architectures take a few minutes each.

All models have the same fully-connected layer depth and width, as well as the same batch size (64) and learning rate (0.0005). The tests ran in Python 3 on an OSX operating system with GPU-support.

4.3 Comparison of Performance

We evaluated four models on the Animals dataset: The original RN used as baseline, an SCHRN with 7 categories, an UCHRN with 8 categories, and the FDHRN with 8 categories. We compared both training times and testing accuracies achieved. The results are summarized in Table 1, with values averaged over several runs and rounded. All models converge by 200 epochs.

Even for merely 25 objects, the training time of the baseline RN is one or two orders of magnitude higher than that of the HRN models. This is expected as, in general, the time complexity of the HRNs scale with m^2 instead of n^2, where m is the number of categories and n the number of objects.

We see that, out of all the models tested, the UCHRN achieves superior accuracy at the end of training and also takes the shortest time to finish training. This is likely because its assumptions are satisfied best by the animals dataset (the questions were posed on all animal attributes). The SCHRN performs worse as the fixed categories given by the dataset (mammal, bug, etc.) are not particularly suited for the

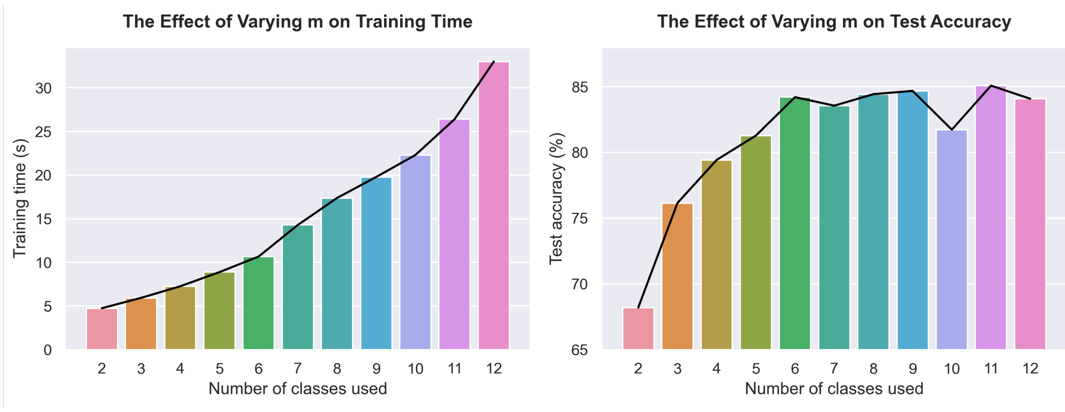

Figure 5: Effect of the number of categories, m, on training time and testing accuracy.

query questions. FDHRN does not perform as well as UCHRN due to its higher variance. The main advantage of FDHRN is that it can infer better categories when many object attributes are irrelevant – but that is not the case in our data.

4.4 Exploring the Optimal Number of Classes

HRNs that support a different number of categories, such as UCHRNs or FDHRNs, allow us to create a large spectrum of models between baseline RNs and simple MLPs. When the number of categories m is the same as the number of objects n, the resulting HRN is effectively a simple RN operating on uncategorized objects, since each object becomes its own category. Whereas when m=1, an HRN resembles more an MLP model that operates on one vector representation of only the objects.

For these experiments we use a subset of the Animals dataset with 20,000 thousand questions (again with an equal train-test split) posed on sets of 80 animals, and used an UCHRN with increasing m. We start with $m = 2$ to keep the network structure consistent. Figure 5 shows both how training time and test accuracy change with m.

We see a clear increase in training time as m increases As the number of categories increases, the number of relations to compute among those categories increases as m^2.

Another point in these results is that the accuracy increases rapidly first and then curves off. Once again this is expected as UCHRNs that can leverage more categories perform better, but only up to a certain point. In this dataset, the network has sufficient information to answer the questions as well as it can with 6 categories.

This last observation is consistent with empirical evidence from human reasoning which suggests that we tend to avoid both overly specific categories and overly generic categories in order to form the smallest possible number of concepts that is sufficient to understand well our environment (Rosch et al., 1976; Markman and Wisniewski, 1997).

5 CONCLUSIONS AND FUTURE WORK

In this paper we drew inspiration from how humans learn relations to explore grouping objects into categories as a plausible improvement to neural relational reasoning. We proposed the Hierarchical Relation Network (HRN) framework, and three architectures of this framework that differ in how they categorize objects. HRNs have comparable accuracy with Relation Networks but they run orders of magnitude faster.

Our work serves as a proof-of-concept and an initial exploration into hierarchical relational reasoning. Indeed, there are many open questions to consider: How can we better represent categories in a way that is more flexible than summing objects? What happens when the objects evenly span the space of their attributes and do not form distinct clusters? Can categories be defined in terms of the task at hand in that case?

We believe that hierarchical object classification and relational learning between categories have great potential to work hand-in-hand towards more practical and general machine intelligence.

REFERENCES

Alexander, P. A., Dumas, D., Grossnickle, E. M., List, A., and Firetto, C. M. (2016). Measuring relational reasoning. *The Journal of Experimental Education*, 84(1):119–151.

Anoop, V., Asharaf, S., and Deepak, P. (2016). Unsupervised concept hierarchy learning: A topic modeling guided approach. *Procedia Computer Science*, 89:386–394.

Fisher, D. (1987). Improving inference through conceptual clustering. *Association for the Advancement of Artificial Intelligence*.

França, M. V. M., Zaverucha, G., and d'Avila Garcez, A. S. (2014). Fast relational learning using bottom clause propositionalization with artificial neural networks. *Mach Learn*, 94.

Kazemi, S. M. and Poole, D. (2018). Relnn: A deep neural model for relational learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Krawczyk, D. C. (2012). The cognition and neuroscience of relational reasoning. *Brain Research*, 1428:13–23.

Learning, U. M. (2020). Zoo animal classification. https://www.kaggle.com/uciml/zoo-animal-classification. Accessed: 2020-11-30.

Markman, A. B. and Wisniewski, E. J. (1997). Similar and different: The differentiation of basic-level categories. *Journal of Experimental Psychology: Learning, Memory, and Cognition*.

Pothos, E. M. and Wills, A. J. (2011). *Formal Approaches In Categorization*. Cambridge University Press.

Rosch, E., Mervis, C. B., Gray, W. D., Johnson, D. M., and Boyes-Braem, P. (1976). Basic objects in natural categories. *Cognitive Psychology*, 8(3):382–439.

Santoro, A., Raposo, D., Barrett, D. G., Malinowski, M., Pascanu, R., Battaglia, P., and Lillicrap, T. (2017). A simple neural network module for relational reasoning. *Neural Information Processing Systems*.

Shanahan, M., Nikiforou, K., Creswell, A., Kaplanis, C., Barrett, D., and Garnelo, M. (2020). An explicitly relational neural network architecture. *arXiv preprint arXiv:1905.10307v4*.

Sourek, G., Aschenbrenner, V., Zelezny, F., and Kuzelka, O. (2015). Lifted relational neural networks. *arXiv preprint arXiv:1508.05128v2*.

Valipour, M. and Wang, Y. (2017). Building semantic hierarchies of formal concepts by deep cognitive machine learning. pages 51–58.

Wang, Y. (2010). On formal and cognitive semantics for semantic computing. *International Journal of Semantic Computing*, 04.

Yeung, C. and Leung, H. (2006). Ontology with likeness and typicality of objects in concepts. *Conceptual Modelling*.