Estimating Quantitative Magnitudes Using Semantic Similarity

Jim Davies[1] and Jonathan Gagné[2]

[1]Institute of Cognitive Science, Carleton University
1125 Colonel By Drive, Ottawa, Ontario, K1S 5B6, Canada
jim@jimdavies.org
[2]Systems Design Engineering Department, University of Waterloo
200 University Avenue West, Waterloo, Ontario, N2L 3G1, Canada
jgagne@uwaterloo.ca

Abstract

We present an AI called Visuo that guesses quantitative visuospatial magnitudes (e.g., heights, lengths) given adjective-noun pairs as input (e.g., "big hat"). It uses a database of tagged images as memory and infers un-experienced magnitudes by analogy with semantically-related concepts in memory. We show that transferring width-height ratios from a semantically-related concept yields significantly lower error rates than using dissimilar concepts when predicting the width-height ratios of novel inputs.

Introduction

Visual instantiation is a cognitive process that generates visual and spatial descriptions. People visually instantiate when reading, designing, and planning. We can describe visual instantiation as having two steps. First, the agent must decide what to visualize. This process generates a structured scene description. Second, it has been argued that, in humans, this structured scene description is used to generate an image or bitmapped depiction (Kosslyn, 1994). The present work contributes to the first step.

Specifically, the problem we address is the estimation of quantitative magnitudes. For example, if a person is asked to imagine a "huge broom," he or she will have no trouble doing so, even if the person has never experienced a broom that had been labeled as "huge" before.

Visuo is a computer program that solves this problem with a low error rate. Briefly, the program receives labeled images to populate its visual memory, and then when called upon to visualize (estimate a quantitative magnitude), it intelligently adapts the most semantically related concept in memory.

We will describe how Visuo works, and then evaluate its output. Our hypothesis is that when estimating quantitative magnitudes of unseen concepts, transfer from semantically similar concepts will produce better results than transferring from semantically dissimilar concepts.

Visuo

We will give a brief overview of Visuo before describing the details of its processing.

Visuo works in two stages: training and visualization. In the training stage, it is trained on a database of labels associated with point clouds in digital images. These point clouds act as a rough segmentation of salient objects found within each image. From the segmentations, Visuo can infer the ratio of width to height for each label. Although Visuo is capable of inferring any spatial attribute that can be quantified, only width to height ratio is demonstrated in this paper because there is a lack of availability of adequate databases from which to train.

During the visualization stage, Visuo takes as input an adjective-noun pair (ANP), and produces, as output, a quantitative estimate of the width-height-ratio (WHR) of the object described by the ANP.

If Visuo has experienced examples of the ANP in the training phase, it simply returns the mean of the ratios observed. If Visuo has not, then it must infer what the ratio is through an analogy with a similar noun in memory.

In the next subsections, we will describe Visuo in detail. Because of our limited space, we will describe only those aspects of Visuo relevant to the current evaluation; see Gagne and Davies (in press) for a full description.

Training Stage

Visuo takes in Adjective-Noun Pairs (ANPs), associated with Width-Height Ratios (WHRs). These data are gleaned from the Peekaboom database. Peekaboom is a game that collects data for research (von Ahn, Liu, & Blum, 2006). The game contains 57,797 images, which have labels associated with each image. These associations are data from a related game, the ESP game (von Ahn & Dabbish, 2004). In total there are 11,810 distinct labels. One player, the "boomer," sees an image and an associated label. His or her job is to click parts of the image, revealing those parts to another, anonymously paired player, the "peeker," whose job it is to guess what label the boomer has in front of him or her. When the peeker successfully types in the

Davies, J. & Gagné, J. (2010). Estimating quantitative magnitudes using semantic similarity. *Conference of the American Association for Artificial Intelligence workshop on Visual Representations and Reasoning* (AAAI-10-VRR), 14--19.

word, both receive points and they move on to the next image.

As a result of this game play, we have access to a large image database with labels associated with respective point clouds. We assume that the point cloud roughly describes the shape of the object, although in some cases it does not (e.g., revealing a face rather than a whole body to make the peeker guess "woman"). Input errors resulting from when this assumption is incorrect further increase our reported errors. Therefore, the reported error rates should be viewed as the upper bounds.

We calculate the height of an object by finding the vertical distance between the highest and lowest points on the *y*-axis, and calculate the width by finding the horizontal distance between the two most distant points on the *x*-axis. The WHR is roughly the width divided by the height.
High WHR objects are wide and thin (e.g., a horizon), medium WHR objects are square or round (e.g., a beach ball), and low WHR objects are tall and thin (e.g., a flagpole).

A subset of the entire database was used since the massive database far exceeded the memory capacity of the computers used. Nine objects were used to create the subset: cat, crow, dog, person, raven, skyscraper, tower, pole, and building. The chosen objects were not selected to reduce error rates, and were chosen arbitrarily. We used the first available 100 instances of each concept.

Processing the input. Each WHR is converted from a crisp number to a distribution over fuzzy numbers (Dubois & Prade, 1987). Rather than storing a single number, a vector of membership values (ranging between 0 and 1) is stored for each fuzzy number set. This was done to emulate the varying spatial sensitivity of receptor cells in the human perceptual system, which produces fuzzy outputs depending on how close to their ideal the stimulus is (Hubel & Weisel, 1965) Visuo was designed to be a model of human thought. Detectors have been found to pick up even high-order visual concepts such as buildings and faces (Krieman, Koch, & Fred, 2000). Behavioral data show that people represent things with graded membership to categories in general (Hampton, 2007). We conjecture that higher-level perceptual detectors in the brain also represent spatial magnitudes and detect relevant stimuli with variable category membership as a function of the firing rates of neural populations. The fuzzy numbers are organized in a logarithmic scale (with the addition of zero), reflecting people's natural tendency to think of numbers logarithmically (Dehaene, Izard, Spelke, & Pica, 2008).

Our collection of magnitude detectors is modeled by a distribution, which has a slot for fifteen points on the (roughly) logarithmic mental unit scale (0, 2, 5, 10, 20, 35, 65, 100, 160, 250, 400, 600, 900, 1350, 1800), as shown in Figure 1. Each input distribution modifies a prototype distribution for the given label. Thus, the prototype for "crow" is a vector describing a distribution over fuzzy numbers. Each number in the vector represents the mean fuzzy membership value for that fuzzy number across all instances observed of that label. The prototype represents the WHR of all crows. Upon experiencing another example of the WHR of a crow, each fuzzy membership number v_{avg} in the distribution is averaged with

$$v_{\text{avg}} = \frac{(n \times v_{\text{old}}) + v_{\text{new}}}{n+1}$$

where v_{old} is the previous value, v_{new} is the new value. Following the calculation of v_{avg}, the count n is increased by 1. Visuo incorporates each new experience of the same category into the prototype. For each fuzzy number in the vector, the prototype represents the mean value of the memberships of all exemplars for the corresponding fuzzy number. In this way, the prototype represents an average of all experiences. Inspired by Rosch (1973), prototypes are memories of general concepts of things, abstracted from specific instances. They represent the family resemblance of a category. For quantitative attributes, this is often interpreted as mean values.

WHR r_j for a particular object j is calculated by

$$r_j = 100 \times \left(\frac{\max(\mathrm{X_j}) - \min(\mathrm{X_j}) + 15}{\max(\mathrm{Y_j}) - \min(\mathrm{Y_j}) + 15} \right)$$

where X_j is the set of the *x*-components of all points labeled with object j, Y_j is the set of the *y*-components of all points labeled with object j, $\max(\mathrm{X}_j)$ and $\min(\mathrm{X}_j)$ are the maximum and minimum values in X_j , and $\max(\mathrm{Y}_j)$ and $\min(\mathrm{Y}_j)$ are the maximum and minimum values in Y_j .

The Peekaboom data does not contain adjectives. We chose to label the top 30% of the distribution "high" WHR, the middle 40% "medium" WHR, and the bottom 30% "low" WHR. Although these percentages were arbitrarily chosen, we do not believe that changing them would affect our results. Visuo creates prototypes for crows in general, as well as separate prototypes for high WHR crows, medium, WHR crows, and low WHR crows. This is important because the meaning of a given use of an adjective depends on the context: a large blimp is much bigger than a large mouse. After all of the input has been collected, Visuo has stored in its memory a prototype for every noun (e.g., "crow") and every ANP (e.g., "high WHR crow").

Visualization Stage

The visualization stage takes an ANP as input and produces an estimated WHR as output. It does this by defuzzifying the prototype with the label matching the ANP. When the appropriate prototype is already in memory, Visuo simply retrieves and uses it. This is the trivial case. When there is not an appropriate prototype in memory, Visuo must create one. The way this imagined prototype is created is the main contribution of this work.

We will describe the process with a running example. Suppose Visuo has experienced many crows and buildings, and many of these instances were also labeled as having high, medium, or low WHR. Suppose also that Visuo has experienced ravens, but these experiences were not labeled with respect to the WHR. When Visuo is called upon to

imagine the WHR of a "high WHR raven," it has no prototype labeled as such. It only has the generic "raven" prototype.

Visuo first searches its memory for the most semantically related term for which there *are* WHR labels, and retrieves its prototype. For example, "raven" is compared to each of the generic versions of the prototypes (e.g., tree, crow, city) using the Wu-Palmer similarity measure (Wu & Palmer, 1994) as implemented in NLTK version (Bird & Loper, 2004) of WordNet (Fellbaum, 1998). The prototypes identified to be most similar are selected as the specific source prototype ("high-WHR crow") and the general source prototype ("crow"). The general target prototype is the noun prototype ("raven") that is to be combined with the *attribute modifier* (discussed below) to create the specific target prototype ("high-WHR raven").

Then it finds a mathematical transformation that turns the distribution of "crow" into the distribution of "high WHR crow."

We view adjectives as functional entities that modify their associated noun phrases. This allows Visuo to use

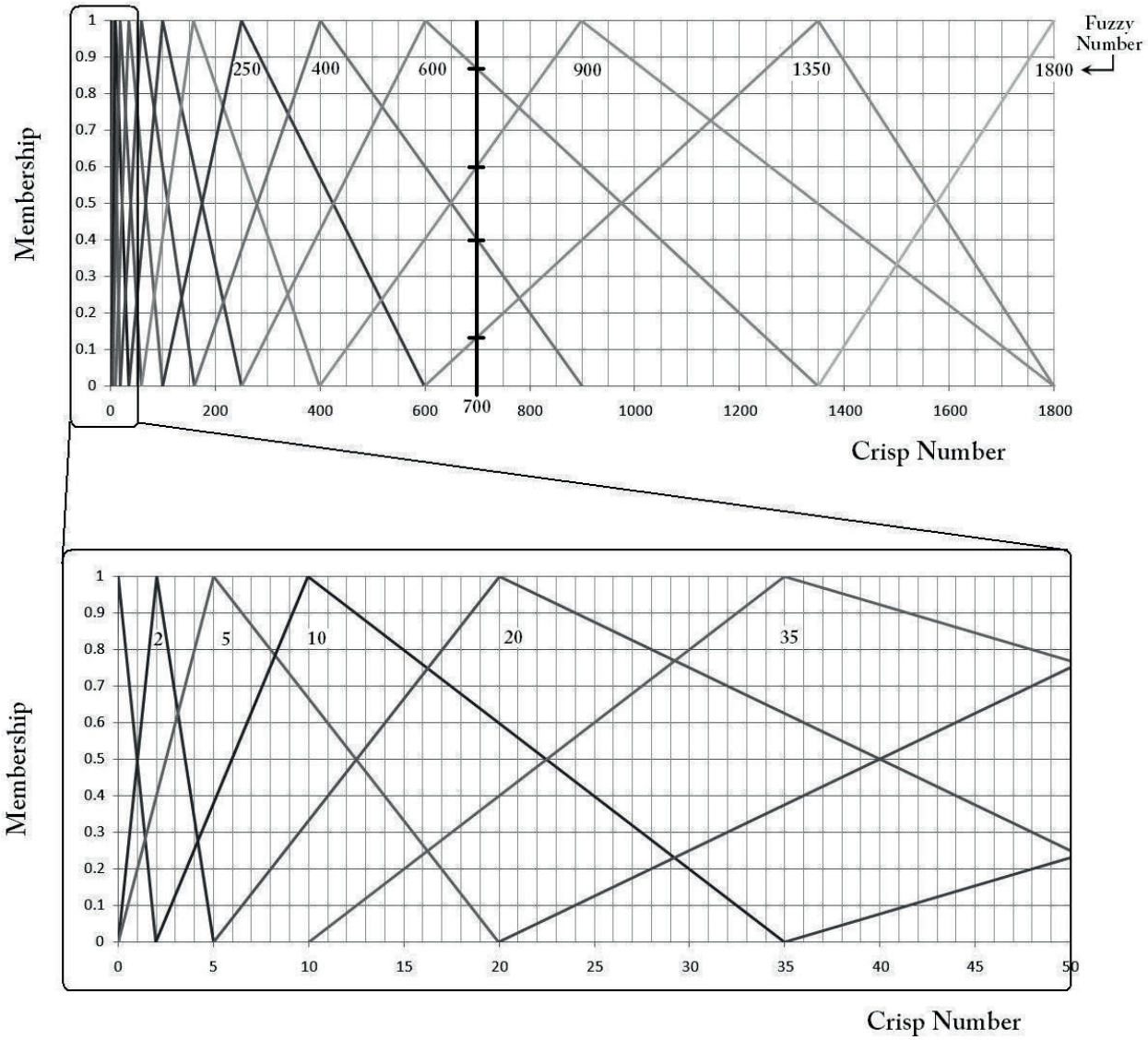

Figure 1. Fuzzy numbers and their actual membership functions at two different scales. The values along the horizontal axis represent crisp number inputs. The values along the top label are the fuzzy number sets. The vertical axis represents the degree of membership of that crisp number in a given set. For example, a crisp input of 700 is a member of fuzzy number 400 to a degree of 0.40, a member of fuzzy number 600 to a degree of 0.8667, a member of fuzzy number 900 to degree 0.60, a member of fuzzy number 1350 to degree 0.1333 and a member of all other fuzzy numbers to degree 0.0.

adjectives to describe new concepts as long as the meaning of the adjective can be abstracted and transferred from a similar situation.

Attribute modifiers are created from adjective senses and used to modify nouns, creating new adjective-noun concepts. They are the instantiation of the functional capabilities of an adjective in a particular context. For example, attribute modifiers are created from the adjective "large" with respect to "crow" and can be applied to "raven" to create a concept for "large raven". Attribute modifiers are data structures that represent how each attribute in a noun prototype is modified by an adjective prototype such that the noun prototype becomes an adjective-noun prototype.

Attribute modifiers contain two distributions, a modifier density and a multiplier. The *modifier density distribution* is a normalized copy of the attribute's distribution of the general source prototype (e.g., "crow"). The *multiplier* is created by a piecewise multiplication of the general source prototype by the specific source prototype. More formally, the multiplier **m** is a vector where each element is defined as

$$\mathbf{m} = \left(\frac{G_0}{S_0}, \frac{G_1}{S_1}, \frac{G_2}{S_2}, \ldots, \frac{G_{k-1}}{S_{k-1}}\right)$$

where k is the size of the distributions, G_i is the ith element in the general source concept distribution, and S_i is the ith element in the specific source concept.

Having found this transformation, it then applies the same transformation to the generic "raven" prototype, resulting in the generation of an imagined "high WHR raven" prototype. If we return to our example, since crows are smaller than ravens, simply applying the concept modifier by multiplying the attribute modifiers' multiplier distribution by the "raven" attribute distributions would result in an inappropriate matching of the numbers in the distribution. In fact, without adjusting the distributions, a "large raven" could have the distribution of what a "small raven" should have, or it could have a distribution consisting of all zeros. Visuo associates the values of the two distributions with the *percent of the distributions that those values cover*.

Visuo creates a density distribution for the modifier values and one for the target prototype, which is a representation of how dense the data is at different parts of the distribution. For example, a density distribution might tell us that most of the data is in the low ranges, with very little in the high ranges. To make this, Visuo copies and normalizes the distribution of the source object's attribute. Normalization is defined here as transforming a distribution such that the sum of the distribution's values equals one. This density distribution is stored with the respective attribute modifier as the modifier density distribution. At this point the multiplier is ready to be combined with the target prototype "raven." The target density distribution is created by normalizing the target distribution.

Each value in the target distribution is multiplied by a percentage of the numbers in the multiplier. This percentage is determined by matching elements of the target density distribution to sections of the multiplier density distribution. For example, the first number in the target distribution might be multiplied by the first two, or even the first 2.6 numbers in the multiplier. If this percentage matching is not done, then large portions of distributions end up being unjustifiably multiplied by zero. The creation of the complete novel concept is now complete.

Once Visuo has a prototype that matches the input, the final step is defuzzification, which is the process of transforming a fuzzy qualitative distribution (such as an attribute's distribution) into a quantitative crisp number. The crisp number N is computed by taking a weighted average of the distribution as defined by:

$$N = \frac{\sum_i \left(u_i \times \mathrm{val}(S_i)\right)}{\sum_i (u_i)}$$

where, u_i is the membership value in the distribution and val(S_i) is the value of the fuzzy number (e.g., val(35F) = 35). The denominator $\Sigma_i(u_i)$ is used to normalize the result.

Visuo defuzzifies from this distribution to create its final output, which is a precise estimation of the WHR of a "high WHR raven."

Evaluation

Our hypothesis is that transferring from a semantically-related prototype will generate more accurate predictions of WHR. In particular, we predict that using semantically-related prototypes will result in better results than transferring from semantically distant prototypes.

To test this hypothesis, we selected nine concepts and tried to predict their WHRs using Visuo. For each concept we created three ANP inputs (e.g., "low-WHR cat," "medium-WHR cat," and "high-WHR cat") and predicted their WHRs using the most similar other concept (of the nine) and the most dissimilar concept. As shown in Table 1, "dog" was the most semantically-related concept to "cat," and "pole" was the most dissimilar.

For each concept, we altered the training phase of Visuo such that each test concept was not associated with any adjectives (e.g., "high WHR"). That is, for predictions about cats, Visuo experienced cat WHRs, but they were not labeled as high or low. For each of the other eight concepts, Visuo experienced not only the WHR, but each WHR was associated with an adjective such as "low WHR." Because of this, Visuo was forced to search memory for a semantically-related concept from which to transfer.

Visualization Phrase	Most Similar	Least Similar	Estimated Ratio (similar)	Estimated Ratio (dissimilar)	Actual Ratio	% Error (similar)	% Error (dissimilar)
Low WHR cat	dog	pole	77.74	66.66	80.58	3.59	18.91
Med WHR cat			112.54	111.74	113.41	0.77	1.48
High WHR cat			168.10	180.64	164.10	2.41	9.60
						avg. 2.26	avg. 10.00
Low WHR crow	raven	pole	78.49	71.68	78.52	0.04	9.11
Med WHR crow			161.23	136.97	136.55	16.58	0.31
High WHR crow			252.37	290.95	284.37	11.92	2.29
						avg. 9.51	avg. 3.90
Low WHR dog	cat	pole	72.69	56.44	64.27	12.30	12.97
Med WHR dog			103.42	99.99	102.33	1.06	2.31
High WHR dog			153.59	174.82	163.46	6.23	6.72
						avg. 6.53	avg. 7.33
Low WHR person	dog	pole	54.67	45.12	54.27	0.73	18.41
Med WHR person			83.65	80.53	83.17	0.58	3.23
High WHR person			129.42	143.48	130.47	0.81	9.50
						avg. 0.71	avg. 10.38
Low WHR raven	crow	pole	80.75	73.63	78.19	3.22	6.01
Med WHR raven			135.03	135.63	147.30	8.69	8.25
High WHR raven			226.14	232.79	211.27	6.80	9.69
						avg. 6.24	avg. 7.98
Low WHR skyscraper	building	cat	38.72	51.00	41.75	7.53	19.95
Med WHR skyscraper			70.67	78.19	73.71	4.21	5.90
High WHR skyscraper			144.45	122.15	137.70	4.78	11.97
						avg. 5.51	avg. 12.60
Low WHR tower	building	cat	28.28	38.37	30.85	8.69	21.73
Med WHR tower			52.46	61.48	53.84	2.60	13.25
High WHR tower			132.08	109.95	127.66	3.40	14.91
						avg. 4.90	avg. 16.63
Low WHR pole	tower	cat	25.02	33.00	18.27	31.19	57.46
Med WHR pole			53.12	77.34	51.45	3.19	40.20
High WHR pole			263.11	222.85	273.13	3.74	20.28
						avg. 12.71	avg. 39.31
Low WHR building	sky-scraper	cat	56.56	71.99	45.18	22.37	45.76
Med WHR building			118.08	127.00	105.68	11.08	18.33
High WHR building			258.38	237.27	285.52	9.98	18.46
						avg. 14.48	avg. 27.52
Overall Average						* 6.98 %	* 15.07 %

Table 1. Visuo's visualization results demonstrating increased accuracy when using semantically similar concepts with respects to semantically dissimilar concepts. * Statistically significant ($p < 0.001$) using Wilcoxon Signed-Rank Test (non-parametric).

Results

The results are presented in Table 1. In the column labeled "Visualization Phrase" are the test ANPs. The columns "most similar" and "least similar" show the terms that the similarity metric determined were the most and least similar to the input phrase (out of the nine concepts under consideration). The "Estimated Ratio (similar)" column shows Visuo's predictions for the WHRs for the test ANP. The column labeled "Estimated Ratio (dissimilar)" shows the WHRs predicted based on the least similar concept. The "Actual Ratio" column shows the correct WHR for the ANP (the mean WHR for the lowest third WHRs of cats). In this test, the accuracy of Visuo is the closeness of the numbers in the Estimated Ratio (similar) and Actual Ratio columns.

The "% Error" columns show the percent error (between predicted and correct) for the similar and dissimilar concepts. If our hypothesis is correct, then the "similar" % Errors should be smaller than the "dissimilar" % Errors.

Transferring data from similar concepts increases accuracy of WHR predictions as compared to using dissimilar concepts. In 24 of the 27 cases, the error was lower for similar than for dissimilar transfers, as predicted. The mean percent errors are significantly different using a Wilcoxon Signed-Rank Test ($Z = 3.61$, $p < 0.001$).

For the similar concepts, Visuo's estimates were very close to the actual values. The worst estimates were "medium crow" (16.58%) and "thick crow" (11.92%),

which used “medium raven” and “thick raven” as a base, respectively. These estimates are quite reasonable considering the Peekaboom database only contained 6 instances of “medium ravens” and 3 instances of “thin ravens” from which to train. Increasing the number of instances dramatically improves results, which can be seen with the width-height ratio prediction for a person. For all three “person” predictions, the error was 0.81% or less. Overall, the average error is 6.98%, which indicates that the cognitive model Visuo can estimate the quantities of unknown spatial attributes with low error rates.

Conclusion

Visuo implements a program that can estimate quantitative values for adjective-noun pairs that have not been experienced by transferring knowledge from prototypes created from experiences of related objects. We tested the accuracy of this method with width-height ratios. The results support our first hypothesis: the average error rate for WHR prediction was low (6.98%).

In addition, we have shown that transferring from semantically-related concepts in memory yields better results. The least semantically-related item provided information that was significantly worse (8.09 percentage points of error different).

Future work will explore the efficacy of the computational choices of using fuzzy numbers, the logarithmic scale, and a larger number of test objects.

References

Bird, S., and Loper, E. 2004. NLTK: The Natural Language Toolkit. *Proceedings of the ACL demonstration session*, Barcelona: Association for Computational Linguistics, pp 214-217.

Dehaene, S., Izard, V., Spelke, E., and Pica, P. 2008. Log or Linear? Distinct Intuitions of the Number Scale in Western and Amazonian Indigene Cultures. *Science* 320 (5880), pp 1217-1220.

Dubois, D., and Prade, H. 1987. Fuzzy numbers: an overview. In: *Analysis of Fuzzy Information Vol. I: Mathematics and Logic.* (J.C. Bezdek, ed.), Boca Raton, Florida: CRC Press.

Fellbaum, C. eds. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Gagné, J. & Davies, J. (in press). Visuo: A model of visuospatial instantiation of quantitative magnitudes. *Knowledge Engineering Review. Special Issue on Visual Reasoning.*

Hampton, J.A. 2007. Typicality, Graded Membership, and Vagueness. *Cognitive Science*, 31 (3), pp 355-384.

Hubel, D.A., and Wiesel, T.N. 1965. Receptive fields and functional architecture in two non-striate visual areas (18 and 19) of the cat. *Journal of Neurophysiology*, 28, pp 229-289.

Kosslyn, S. M. 1994. *Image and Brain: The Resolution to the Imagery Debate.* MIT Press.

Kreiman, G., Koch, C., & Fried, I. 2000. Category-specific visual responses of single neurons in the human medial temporal lobe. *Nature Neuroscience, 3*, pp 946-953

Rosch, E.H. 1973. Natural categories. *Cognitive Psychology*, 4, pp 328-350.

von Ahn, L. and Dabbish, L. 2004. Labeling images with a computer game. *Proceeding of CHI 2004*, pp 319–326.

von Ahn, L., Liu, R., and Blum, M. 2006. Peekaboom: A Game for Locating Objects in Images. *Proceedings of CHI 2006*, pp 55-64.

Wu, Z., and Palmer, M. 1994. Verb semantics and lexical selection. In: *32nd Annual Meeting of the Association for Computational Linguistics*, pp 133-138.