"Look, some green circles!": Learning to quantify from images *

Ionut Sorodoc and **Angeliki Lazaridou** and **Gemma Boleda**
Aurélie Herbelot and **Sandro Pezzelle** and **Raffaella Bernardi**
CIMeC - Center for Mind/Brain Sciences, University of Trento
first.lastname@unitn.it

Abstract

In this paper, we investigate whether a neural network model can learn the meaning of natural language quantifiers (*no, some* and *all*) from their use in visual contexts. We show that memory networks perform well in this task, and that explicit counting is not necessary to the system's performance, supporting psycholinguistic evidence on the acquisition of quantifiers.

1 Introduction

Multimodal representations of meaning have recently gained a lot of attention in the computational semantics literature. It has been shown, in particular, that the meaning of content words can be modelled in a cognitively – and even neuroscientifically – plausible way by learning representations from both the linguistic and visual contexts in which a lexical item has been observed (Anderson et al., 2013; Lazaridou et al., 2015). Such work has been crucial to advance the development of both a) a computational theory of meaning rooted in situated language use, as pursued by the field of Distributional Semantics (Clark, 2012; Erk, 2012) and b) vision-based applications such as image caption generation and visual question answering (Antol et al., 2015), going towards genuine image understanding.

Both distributional semantics and visual applications, however, struggle with providing plausible representations for function words. This has theoretical and practical consequences. On the theoretical side, it simply reduces the explanatory power of the model, in particular with respect to accounting for the compositionality of language. On the practical side, current vision systems are forced to rely on background language models instead of truly interpreting the words of a query or caption in the given visual context. As a consequence, if e.g. the sentence *I see some cats* is more frequent than *I see no cat*, language model-based applications will tend to generate the first even when the second would be more appropriate.

In this paper, we start remedying this situation by investigating one important class of function words: natural language quantifiers (e.g. *no, some, all*). Quantifiers are an emerging field of research in distributional semantics (Grefenstette, 2013; Herbelot and Vecchi, 2015) and, so far, haven't been studied in relation with visual data and grounding. We make a first step in this direction by asking whether the meaning of quantifier words can be learnt by observing their use in the presence of visual information. We observe that in grounded contexts, children learn to make quantification estimates before being able to count (Feigenson et al., 2004; Mazzocco et al., 2011), using their Approximate Number Sense (ANS). We ask whether Neural Networks (NNs) can model this ability, and we evaluate several neural network models, with and without numerical processing ability, on the task of matching a non-cardinal to a referent in a grounded situation.

NNs have been shown to perform well in tasks related to quantification, from counting to simulating the ANS. Seguí et al. (2015), for instance, explore the task of counting occurrences of an object in an image using convolutional NNs, and demonstrate that object identification can be learnt as a surrogate of counting. Stoianov and Zorzi (2012) show that the ANS emerges as a statistical property of images in deep networks that learn a hi-

*This project has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 655577 (LOVe); ERC 2011 Starting Independent Research Grant n. 283554 (COMPOSES). We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used in our research.

erarchical generative model of visual input. To our knowledge, however, there hasn't been any attempt so far to model the use of non-cardinals (*no, some, all*) in a visual quantification task.

Our paper builds on previous work by proposing a NN model of quantifier terms which can be related to the acquisition of the ANS, with two main contributions: First, we propose a novel experimental setup in which, given a set of objects with different properties (e.g., circles of different colors), the model learns to apply the correct quantifier to the situation (e.g. *no, some, all* circles are red). Second, we show that, as observed in children, our best model does not need to be able to count in order to quantify.[1]

2 Visual Quantification Dataset

Linguistic quantifiers and their logical properties have been a major object of study in the field of formal semantics since its inception (Montague, 1974). It is posited that, in an example such as *some circles are green*, the quantifier (*some*) expresses a relation between a domain restrictor (*circles*) and the quantifier's scope (*are green*). In this paper, we fix the domain and focus on the scope: We ask whether, given an image with objects from a single domain (circles), a model can learn to globally quantify the objects with a certain property, deciding whether *all*, *some*, or *no* circles have that property. Here, we use color as an example property to quantify over.

Images. In order to focus on the quantification task, barring out any effect from data preprocessing, we create an artificial dataset with clear visual properties (see below). Our dataset consists of images with 1 to 16 circles of 15 different colors, and we generate all possible combinations of different numbers of circles (from 1 to 16) with all possible combinations of colors. Figure 1 presents one of the images in the dataset.

Image representation. In order to avoid effects from visual pre-processing, the dataset is presented to the quantification network with (automatically produced) gold standard information about image segmentation and object identification. That is, the network knows *where* objects are, and *what* they are (circles of different, easily identifiable colors). Concretely, we represent each picture as a set of up to 16 circles (e.g. Figure 1) placed in 16 fixed image cells. Furthermore, we associate each of the circle-color combination with real-valued vectors of dimensionality 20 that are normalized to unit norm. All circles are identical in shape and size, so the differences observable in the vector representations can be taken to express the color property of the objects. We ensure that the dataset does not include 'confusable' objects by further constraining the vectors to have low pairwise similarity.[2] On the other hand, to prevent overfitting, we add a small amount of noise to all vectors, generated for each dimension from a Gaussian distribution with mean 0 and variance 1/5 of the original variance of that dimension. Intuitively, the Gaussian noise simulates natural variations in a given property, e.g., two tennis balls being of slightly different shades of yellow. This is applied to both training and test data. Finally, our images may contain *empty cells*, viz. parts of the image with no object in it (e.g., in Figure 1 there are 5 empty cells.) These are similarly represented by a vector, randomly generated so that it be orthogonal to all the other object vectors.

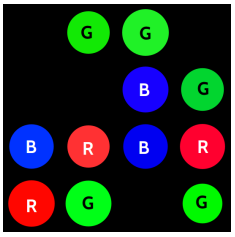

Figure 1: One of the images in our visual quantification dataset. Letters indicate color: R(ed), G(reen), B(lue).

Queries Each image in the dataset is associated with a *query*, i.e., the property we want to quantify over, and the task of the model is to associate the correct *quantifier* with the query for the image. For instance, the query associated to the image in Figure 1 is *green* and the correct quantifier is *some*. *Some* encodes "at least one but not all circles have color X", *all* encodes "all circles have color X" and *no* "no circle has color X". Our dataset contains 5K <image, query, quantifier> datapoints split equally amongst the three quantifiers,[3] which will be used to evaluate our models.

[1] Our code and data are available at `https://github.com/sorodoc/quantifiers_torch`.

[2] We fix this parameter to values not exceeding a cosine similarity of 0.7

[3] Note that, although the *all* quantifier generates fewer images than *no* and *some*, it is possible to create balanced data by producing noisy variations of a same image.

3 Models

Our aim is to understand whether NNs can learn to quantify objects of a certain property in a given image. Our main hypothesis in this paper is that for acquiring such ability the model does not need to rely on exact number information but it can do so by computing the gist of the queried property in the image, thus simulating the human ANS. We build three models to test this hypothesis.

Quantification Memory Network (qMN): This is the model we propose in this paper; it is designed to show that knowing how to count is not a *necessary* condition to be able to learn to quantify. It is an adaptation of the memory network of Sukhbaatar et al. (2015) for visual quantification. As shown in Figure 2, the model consists of a memory with 16 slots, one for each image cell. It computes the dot product between each memory slot and the vector query, obtaining 16 scores, which are then fed into a softmax classifier to derive a valid probability distribution. These normalized scores are used to derive the "gist" of the image (a 20-D vector), by computing a weighted sum over cell vectors in the memory slots, where the weights are taken from the probability distribution that is output by the softmax classifier. Finally, a non-linear transformation with a ReLU activation is applied over the concatenation of the "gist" and query vectors. The vector dimensionality is reduced to 3 by linear transformation and a softmax classifier is applied on top of that, deriving a probability distribution over the three quantifiers. The "gist" vector is an aggregate of the memory, and information about individual objects is lost, such that the model is not able to count. However, the similarity between the "gist" and the query reflects the ratio (rather than the exact number) of objects of that color in the image. To make this explicit, in the case of 'all' , the gist and query vectors will be almost identical, in the case of 'no' there will hardly be any trace of the query in the gist, making them different, and in the case of 'some' query and gist will be somewhat similar.

Counting model: We note that a simple rule-based model comparing the cardinalities of the restrictor and scope in the query would achieve 100% accuracy. But we want to check to what extent a NN model based on softmax and non-linear transformation, similar to qMN, can learn to quantify when provided with *exact number information* about the objects and their colors. Indeed, despite the obvious logical interpretation of quantifiers as ratios between two magnitudes, it is unclear whether this logical operation is easily learnable in a visual connectionist model. In this setup, we build for each image a 16-D feature vector, one dimension for each of the 15 colors plus one for the empty cell. To each dimension we assign a value encoding the frequency of the color in the image scaled by the similarity of that color to the query (recall that, because of the added Gaussian noise, a given yellow circle may not be identical to the query *yellow*). This way, the quantity of objects of a given color is encoded in the dimensions of the vector as if the model was counting. The query is represented by a one-hot 16-D vector, encoding the color the model is asked to quantify over. The feature and query vectors are concatenated. As in the qMN model, we then apply a linear transformation followed by a ReLU activation and a softmax classifier.

Recurrent Neural Network (RNN): As an alternative model with a visual memory, we also implement an RNN that uses the hidden state to encode information about the image's gist. At each timestep, the RNN receives as input first the query vector followed by each of the 16 objects vectors. At the last timestep, the hidden layer is fed to a linear transformation, reducing its size to 3, on top of which a softmax classifier is applied to obtain a probability distribution over the quantifiers. As opposed to the qMN, the RNN does not explicitly model the similarity between the query and the color of the objects in the image.

All models are trained with cross-entropy to predict the correct quantifier.

4 Experimental setup

We randomly divide the 5K data points into training, validation and test set (70%, 10% and 20%). We test the models in 3 experimental setups. The first setup, **familiar**, is the simplest, and tests whether models are able to quantify previously observed ("familiar") colors and quantities. In the **unseen quantities** setup, we create training and test sets so that there is no overlap with respect to the number of objects in the image: 4, 9 or 13 objects are used at test time and all other quantities at training/validation time (i.e., 1-3, 5-8, 10-12, 14-16). Finally, in the **unseen colors** setup, we make sure training and test sets differ with respect to objects' color: The models are trained/validated on

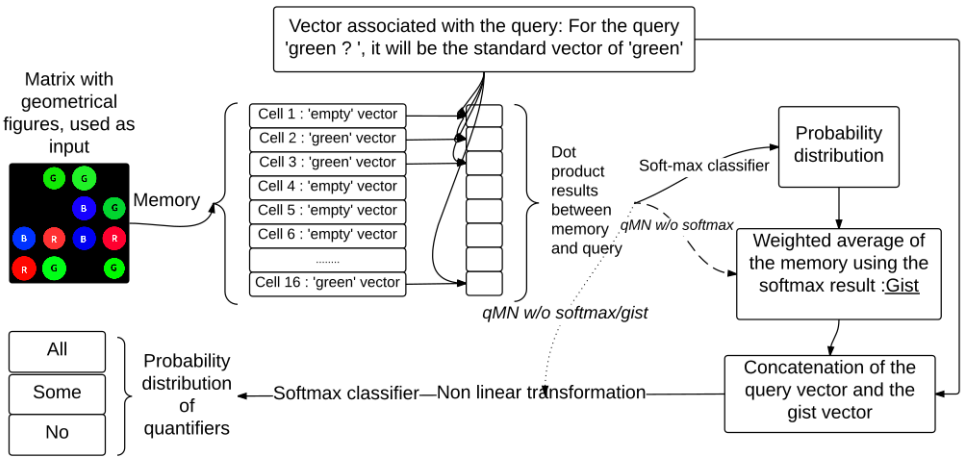

Figure 2: Quantification Memory Network model

Models	**familiar**	**unseen quantities**	**unseen colors**
RNN	65.7	62.0	49.7
Counting	86.5	78.4	32.8
qMN	**88.8**	**97.0**	**54.9**
-softmax	85.9	66.6	54.4
-softmax/gist	51.4	51.8	44.4

Table 1: Model accuracies (in %).

10 colors and tested on 5 additional, unseen colors. We expect that the use of the gist in our model, which implements *global* quantification over objects of a certain property, will allow it to generalize well when tested against unseen quantities.

5 Results

As shown in Table 1, having exact number information is not necessary for learning to quantify: The qMN model, which does not explicitly count, is more accurate than the Counting model in all test conditions. Even though both models outperform the RNN model when tested on unseen number of objects, only the qMN model truly generalizes the learnt quantification operation. The performance of all models drops when tested on unseen colors, though qMN still performs best and the decrease in performance in Counting is much worse than in the qMN model (-53.7 vs. -34). Lines "-softmax" and "-softmax/gist" in Table 1 show that both the softmax and the "gist" are crucial elements of the model; removing them causes significant performance drops in all test conditions.

By looking at the confusion matrices for the qMN model we observe that there is generally more confusion between *no* and *some* than in pairs involving *all*; the gist for *some* is an average of potentially several different colors, and thus less straightforwardly interpretable. In the 'familiar' test, most of the errors come from situations in which the model confused "some" with "no" *and* the image contains just 1 or at most 2 occurrences of the queried color. Hence, the increase in performance from the familiar to the unseen quantity test (+8.2) is due to the absence of very small cardinalities in the image (the lowest is 4 items.) As for *all*, in both the 'familiar' and the 'unseen quantities' conditions it's nearly always classified correctly. This is to be expected because in this case, the "gist" computation produces a vector which should be cleanly equivalent to the query (minus the effect of noise). When moving to unseen properties performance decreases, indicating that the network might have overfitted to the particular colors in the training set. Although we'll need to address this behaviour in further work, we don't consider it a weakness of a *quantification* model per se: the problem to be solved is one of object/property recognition and not of quantification.

6 Conclusion

We have shown that a memory network can learn to quantify objects of a certain property, given some visually grounded training data involving small sets. Given that the number of memory cells is parametric, the model should in principle be able to scale to much larger number of cells. Our future work will focus on modelling the entire quantifier meaning, varying not only the quantifier scope but also its *restriction*.

References

Andrew J Anderson, Elia Bruni, Ulisse Bordignon, Massimo Poesio, and Marco Baroni. 2013. Of words, eyes and brains: Correlating image-based distributional semantic models with neural representations of concepts. In *EMNLP*, pages 1960–1970.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433.

Stephen Clark. 2012. Vector space models of lexical meaning. In Shalom Lappin and Chris Fox, editors, *Handbook of Contemporary Semantics – second edition*. Wiley-Blackwell.

Katrin Erk. 2012. Vector space models of word meaning and phrase meaning: a survey. *Language and Linguistics Compass*, 6:635–653.

Lisa Feigenson, Stanislas Dehaene, and Elizabeth Spelke. 2004. Core systems of number. *Trends in cognitive sciences*, 8(7):307–314.

Edward Grefenstette. 2013. Towards a formal distributional semantics: Simulating logical calculi with tensors. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics (STARSEM)*.

Aurélie Herbelot and Eva Maria Vecchi. 2015. Building a shared world: Mapping distributional to model-theoretic semantic spaces. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal. https://www.aclweb.org/anthology/W/W13/W13-0204.pdf.

Angeliki Lazaridou, Nghia The Pham, and Marco Baroni. 2015. Combining language and vision with a multimodal skip-gram model. In *Proceedings of NAACL*.

Michèle MM Mazzocco, Lisa Feigenson, and Justin Halberda. 2011. Preschoolers' precision of the approximate number system predicts later school mathematics performance. *PLoS one*, 6(9):e23749.

Richard Montague. 1974. The proper treatment of quantification in ordinary English. In R. Thomason, editor, *Formal Philosophy*, pages 247–270. Yale University Press, New Haven.

Santi Seguí, Oriol Pujol, and Jordi Vitria. 2015. Learning to count with deep object features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 90–96.

Ivilin Stoianov and Marco Zorzi. 2012. Emergence of a'visual number sense'in hierarchical generative models. *Nature neuroscience*, 15(2):194–196.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. http://arxiv.org/abs/1503.08895.