Abstract

This work aims at modeling how the meaning of gradable adjectives of size (‘big’, ‘small’) can be learned from visually-grounded contexts. Inspired by cognitive and linguistic evidence showing that the use of these expressions relies on setting a threshold that is dependent on a specific context, we investigate the ability of multi-modal models in assessing whether an object is ‘big’ or ‘small’ in a given visual scene. In contrast with the standard computational approach that simplistically treats gradable adjectives as ‘fixed’ attributes, we pose the problem as relational: to be successful, a model has to consider the full visual context. By means of four main tasks, we show that state-of-the-art models (but not a relatively strong baseline) can learn the function subtending the meaning of size adjectives, though their performance is found to decrease while moving from simple to more complex tasks. Crucially, models fail in developing abstract representations of gradable adjectives that can be used compositionally.

1 Introduction

There is no doubt that planets are big things. Among the planets of our Solar System, however, Mars is unquestionably a small planet (though not the smallest), while Saturn is definitely a big one (though not the biggest). This example highlights some crucial properties of gradable adjectives (hence, GAs). First, what counts as ‘big’ or ‘small’ is relative, i.e., determined by the context: Phobos is both a big moon of Mars and a small celestial body of the Solar System. This makes GAs different from non-gradable or absolute adjectives like ‘open’, ‘empty’, ‘red’: Mars, for example, is red and rocky in any circumstance. More formally, the compositional semantic properties of GAs are subsective since they select a subset of entities denoted by the noun they modify, which acts as a reference set (\(|\text{big}| \subseteq |\text{moon}|\)), while non-GAs are intersective (\(|\text{Galilean}| \cap |\text{moon}|\)). This has consequences for the inferences they license: if Ganymede is both a Galilean moon and a celestial body, we can infer that Ganymede is a Galilean celestial body. In contrast, if it is a big moon and a celestial body, the inference that Ganymede is a big celestial body is not valid (Partee, 1995).

Second, besides depending on a contextually-given reference set, GAs rely on orderings, i.e., they denote functions that map entities onto scales of degrees (Cresswell, 1976; Kennedy, 1999). Using ‘big’ or ‘small’, thus, implies mapping a target object onto a size scale, which allows us to use degree morphology to express that Saturn is bigger than Mars (comparative form) or that Mercury is the smallest planet (superlative form). As for the non-inflected, so-called positive form of GAs (e.g., Saturn is a big planet), its interpretation involves applying a statistical function that makes use of a standard threshold degree (Kamp, 1975; Pinkal, 1979; Barker, 2002; Kennedy, 2007).

Third, GAs are considered to be vague, because whether they apply or not to a given entity can be a matter of debate among speakers (Van Deemter, 2012; Lassiter and Goodman, 2017). Since people might rely on slightly different functions involving probabilistic thresholds, there are often borderline cases: e.g., Neptune (i.e., the fourth planet out of eight in terms of size) could be considered as a big planet by most but not all speakers in all situations.

Our aim in this work is to computationally learn the meaning of size GAs (‘big’, ‘small’) from visually-grounded contexts. Based on the semantic properties of such expressions (context-dependence, statistically-defined interpretation, vagueness), we tackle the task as a relational problem in the domain of visual reasoning (similarly, e.g., to spatial problems like assessing whether ‘X is on the left of Y’). Simply put, a model needs
to consider the entire visual context (not just the queried object) in order to solve the task. Such setup resembles experimental paradigms in developmental psychology which test how children interpret GAs when applied to objects grounded in visual scenes (Barner and Snedeker, 2008). Evidence shows that children learn to use GAs compositionally early on: when asked to assess whether an object is ‘tall’ or ‘short’ in a visual context, 4-year-old children are able to (a) restrict the reference set by means of linguistic cues and (b) derive a tall/short threshold relative to that set. That is, 4-year-olds do not interpret GAs categorically but compositionally. This is radically different from how adjectives are treated in current visual reasoning approaches, which consider them static labels standing for attributes (size, color, material) whose value is fixed across contexts (Johnson et al., 2017a; Santoro et al., 2017): i.e., for current models, Saturn is always big and Mars is always small.

To model GAs in a relational fashion, we rely on a statistical function that is found to be best predictive of human interpretations (Schmidt et al., 2009) and build MALeViC, a battery of datasets for Modeling Adjectives Leveraging Visual Contexts (see Figure 1). Each dataset, including 20K synthetic visual scenes and automatically-generated language descriptions (e.g., ‘the red square is a big square’), is used to test different abilities. We experiment with several models and show that FiLM (Perez et al., 2018) and, to a lesser extent, Stacked Attention Networks (Yang et al., 2016) can learn the function subtending the meaning of size adjectives, though their performance is found to decrease while moving from simple to more complex tasks. Crucially, all models fail in developing abstract representations of gradable adjectives that can be used compositionally.

2 Related Work

Computational Linguistics  Computational approaches to GAs have mostly focused on automatically ordering elements with respect to their intensity (e.g., good<great<excellent, de Marneffe et al., 2010) to overcome a problem with lexical resources like WordNet (Fellbaum, 1998), which consider words like ‘small’ and ‘minuscule’ as synonyms. These efforts showed the potential of using techniques based on word embeddings (Kim and de Marneffe, 2013), web-scale data (Sheinman et al., 2013; De Melo and Bansal, 2013), or their combination (Shivade et al., 2015; Kim et al., 2016) to determine the relative intensity of different words on a scale. By focusing on the ordering between adjectives, however, these works do not shed light on how the meaning of such expressions is determined by contextual information. This goal, in contrast, has been pursued by work on automatic Generation of Referring Expressions (GRE), where GAs have represented an interesting test case precisely due to their context-dependent status (van Deemter, 2006). Several small datasets of simple visual scenes and corresponding descriptions have been proposed (van Deemter et al., 2006; Viethen and Dale, 2008, 2011; Mitchell et al., 2010, 2013a), and forerunner GRE algorithms have been tested to refer to visible objects using attributes like size (Mitchell et al., 2011, 2013b). Due to their extremely small size, however, these datasets are not suitable for large-scale deep learning investigations.

Language & Vision Recent work in language and vision has dealt with (gradable) adjectives from at least three perspectives. First, multi-modal information has been used to order real-world entities with respect to an attribute described by a GA. Bagherinezhad et al. (2016), for example, assess whether elephants are bigger than butterflies.
Thus, rather than modeling the meaning of GAs, this work focuses on the relative size of object types, which is crucially different from our goal. Second, visual gradable attributes have been used as fine-grained features to answer questions about entities in natural scenes (Antol et al., 2015; Hudson and Manning, 2019), to perform discriminative and pragmatically-informative image captioning (Vedantam et al., 2017; Cohn-Gordon et al., 2018), and to discriminate between similar images in reference games (Su et al., 2017). In these works, however, GAs are labels standing for fixed attributes: e.g., an airplane is annotated as ‘large’ or ‘small’ with no or little relation to the visual context. Third, gradable attributes are explored in work on visual reasoning (Johnson et al., 2017a; Yi et al., 2018), where multi-modal models are challenged to answer complex, high-level questions about objects and their relations in synthetic scenes. Surprisingly, however, questions involving GAs are treated as non-relational: e.g., a given size always corresponds to either ‘big’ or ‘small’, regardless of the objects in the visual context. One exception is Kuhnle et al. (2018), whose investigation of superlative (‘biggest’) and comparative (‘darker’) GAs represents a first step toward a relational account of these expressions. To our knowledge, we are the first to tackle the modeling of positive GAs as a linguistically- and cognitively-inspired relational problem in language and vision.

3 Tasks

To test model abilities in learning size GAs, we propose 4 main tasks (see Figure 2). All tasks are framed as a sentence-verification problem: given an image depicting colored, geometric shapes and a sentence about one object’s relative size, systems are asked to verify whether the given sentence is true or false in that context. These tasks are aimed at assessing one ability at a time, and are specifically designed to form a pipeline of increasingly difficult operations. Tasks differ with respect to:

- the statistical function at play: a max/min function (SUP) or a threshold function (POS);
- the number of geometric shape types: one (SUP₁/POS₁) or several;
- the scope of the reference set: the entire visual scene or a subset of objects (SET; only applicable with several shape types per scene).

SUP₁ This task tests whether a model is able to interpret size GAs in their superlative form: e.g., ‘The yellow triangle is the biggest triangle’ (only triangles in the scene). To solve this task, a model is required to (1) identify the queried object, (2) measure object size, and (3) determine whether the target object has the largest or smallest size in the entire visual scene.

POS₁ This task evaluates the ability of models to interpret positive-form size GAs: e.g., ‘The white square is a small square’ (only squares in the scene). To adequately solve this task, a model is required to (1) identify the queried object, (2) measure object size, and (3) determine whether the target object has the largest or smallest size in the entire visual scene.

POS This task is an extension of POS₁ where the restriction to one single geometric shape type...
per scene is lifted: e.g., ‘The red circle is a big object’ (any shape in the scene) – see Figure 2. As before, and in contrast to the next task, the reference set is the entire visual scene.

**SET+POS** Finally, this task assesses the ability of models to interpret positive-form GAs with respect to a restricted context: e.g., ‘The white rectangle is a big rectangle’ (any shape in the scene). To solve this task, in addition to the skills required to address the POS task, a model needs to determine the relevant reference set (e.g., the set of rectangles in the scene) and to apply all POS operations to this set. This task, thus, brings together all the key components that make up the semantics of size adjectives, as described in Section 1.

### 4 Datasets

**Visual Data** For each task, we build a dataset of synthetic scenes (1478 x 1478 pixels) depicting 5 to 9 colored objects\(^2\) on a black background. Each object is randomly assigned one shape among circle, rectangle, square, triangle; one color among red, blue, white, yellow, green; one area among 10 predefined ones (based on number of pixels) that we label using tens ranging from 30 to 120 (i.e. 30, 40, 50, ... 120); and a given spatial position in the scene which avoids overlapping with other objects. In tasks SUP\(_1\) and POS\(_1\) only one shape type is present in a given scene, while in tasks POS and SET+POS several shape types are present.

**Linguistic Data** While generating the visual data, ground-truth labels (‘biggest’, ‘smallest’, ‘big’, ‘small’) are automatically assigned to each object based on the area of the objects present in the reference set. For tasks POS\(_1\), POS and SET+POS, for each object in the scene the generation algorithm generates a sentence based on the template: *The <color> <shape> is a <size> <shape>.* For task SUP\(_1\), for objects that are biggest/smallest in the scene the algorithm generates a sentence using the template *The <color> <shape> is the <sup size> <shape>.* In order for a sentence to be licensed, the following constraints have to be met: (1) The queried object is uniquely identifiable within the scene, either by <color> in tasks SUP\(_1\) and POS\(_1\) or by <color, shape> in tasks POS and SET+POS (this ensures there is no ambiguity regarding the target); (2) the area of the queried object must be in the 40-110 range, so to avoid querying objects whose size would always be smallest/biggest in any scene irrespective of the area of the other objects (see Figure 3, which shows the distribution of queried objects’ areas per question type in the train set of POS; as can be seen, no objects with an area equal to either 30 or 120 are present). In task SET+POS, we include two additional constraints: (3) the queried object is neither the biggest nor the smallest object in the entire scene (i.e., the max/min function is not effective without identifying the reference subset); and (4) there are at least three objects in the scene with the same shape as the target (i.e., the reference set includes at least three objects), so to make the computation of the threshold function required to solve the task. Only sentences meeting these constraints are generated and the corresponding scene retained. For each true sentence, a false sentence is automatically generated by replacing the true adjective (e.g., ‘big’) with the false one (e.g., ‘small’).

**Threshold Function** To assign ground-truth labels to objects in tasks POS\(_1\), POS and SET+POS, we make use of a threshold function experimentally determined by Schmidt et al. (2009), who explore a number of statistical functions to describe speakers’ use of the adjective ‘tall’ (following the authors, we assume these functions to be valid for any GAs, including ‘big’).\(^3\) In particular, we use their Relative Height by Range (RH-R) function, noted also that, while size may be argued to be two-dimensional, in our approach we treat it as one-dimensional (i.e., based on number of pixels), similarly to tallness.

---

\(^2\)This resembles the setup used by Barner and Snedeker (2008) in their first experiment, which involves 9 objects.

\(^3\)This resembles the setup used by Barner and Snedeker (2008) in their first experiment, which involves 9 objects.
according to which any item within the top $k\%$ of the range of sizes in the reference set is ‘tall’ ('big'). According to this function, the threshold $T$ for a given context $C$ is defined as follows: $T(C) = \text{Max} - k \cdot (\text{Max} - \text{Min}),$ where $k \in [0,1]$, Max is the maximum size in $C$, and Min the minimum size (see Figure 1). In our data, we make the simplifying assumption to consider ‘small’ any object that is not ‘big’. This way, we also avoid dealing with negative statements. Since Schmidt et al. (2009) experimentally show $k = 0.29$ to best fit their human data, we use this value as our reference $k$. To account for vagueness, for each scene we randomly sample a $k$ from the normal distribution of values centered on 0.29 ($\mu = 0.29, \sigma = 0.066$),$^4$ as illustrated in Figure 1. This introduces some perturbation in the definition of $T$, which mimics the fact that speakers may rely on slightly different definitions of GAs (i.e., what counts as ‘big’ for one person might not always count as ‘big’ for another one; Raffman, 1996; Shapiro, 2006; Sharp et al., 2018), with communication still being successful in most of the cases.

Datasets The four final datasets, including 20K datapoints each, are perfectly balanced with respect to the number of cases for each combination of variables used. In particular, 250 <scene, sentence> datapoints are included for each of the 80 classes (4 shapes * 5 colors * 2 sizes * 2 ground truths), where one class is e.g.: <red> <circle> <big> <true>. Such balancing is kept when randomly splitting the datasets into train (16K cases), validation (2K) and test (2K) sets.

5 Models

We test 3 models that have proved effective in visual reasoning tasks (Johnson et al., 2017a; Suhr et al., 2018; Yi et al., 2018). All models are multi-modal, i.e., they use both a visual representation of the scene and a linguistic representation of the sentence. Unless otherwise specified, visual features representing the scenes are extracted via a fixed Convolutional Neural Network (CNN) fed with images resized to 224 x 224 pixels. In particular, features from the conv4 layer of a ResNet-101 (He et al., 2016) pre-trained on ImageNet (Russakovsky et al., 2015) are used.

CNN+LSTM This model simply concatenates the CNN visual features with a representation of the sentence encoded using the final hidden state of an LSTM (Hochreiter and Schmidhuber, 1997). These concatenated features are passed to a Multi-Layer Perceptron (MLP) that predicts the answer.

Stacked Attention (CNN+LSTM+SA) This model combines the CNN visual features and the LSTM final state by means of two rounds of soft spatial attention. An MLP over the combined representation predicts the answer (Yang et al., 2016).

CNN+GRU+Feature-wise Linear Modulation (FiLM) This model (~100% on CLEVR; Johnson et al., 2017a) processes the image features by means of 4 residual blocks where the visual representation is linearly transformed by the sentence embedding (i.e., the final hidden state of a GRU; Chung et al., 2014). After a global max-pooling, a two-layer MLP classifier outputs a softmax distribution over the answers (Perez et al., 2018).$^5$

Experimental Setup For each model in each task, we experiment with 16 configurations of hyper-parameters, i.e., 4 learning rates (5e-5, 3e-4, 5e-4, 5e-3) * 2 dropout values (0, 0.5) * 2 batch normalization options (yes, no). Each model configuration is trained 3 times with randomly initialized weights for 10K iterations (40 epochs), and the best configuration is chosen based on average validation accuracy. In total, 576 models (3 architectures * 4 tasks * 48 runs) are tested. For comparison with previous work (Perez et al., 2018; Kuhnle et al., 2018), we also test FiLM end-to-end (i.e., trained from raw pixels) with the same hyper-parameters as above, for a total of 192 models (4 tasks * 48 runs). Since no substantial differences are observed between the two versions (in line with Perez et al., 2018), we focus on the results by the less tailored and less computationally-expensive models using pre-trained features.

6 Results

Overall, we observe the expected pattern of results, in line with the conjectured, increasing difficulty of the tasks: accuracy declines from SUP1 to POS1 and from POS to SET+POS. There are, however, clear differences across models.

$^4$By setting $\sigma = 0.066$, we expect 99.7% $k$ values to be $\pm 3$ standard deviations from 0.29, i.e., ranging from 0.09 to 0.49; 95% $k$ within $\pm 2$ SD (0.16-0.42); 68% within $\pm 1$ SD (0.22-0.36). $\sigma$ value was set experimentally to keep $k < 0.5$.

$^5$For CNN+LSTM and CNN+LSTM+SA, we use the implementations by Johnson et al. (2017b). For FiLM, we use the implementation by Perez et al. (2018). All models are trained using Python 3.5.2 and PyTorch v1.0.1.
CNN+LSTM does well on SUP (the simplest task) but performs around chance on all other tasks (which require a threshold function). Both CNN+LSTM+SA and FiLM obtain high accuracy on POS₁/POS, while FiLM neatly outperforms CNN+LSTM+SA in the most challenging SET+POS task, proving its higher ability to reason over complex, context-dependent linguistic expressions. Accuracy scores are reported in Table 1. In the following, we describe the results per task in more detail.

**SUP₁** Both CNN+LSTM+SA and FiLM obtain perfect accuracy (~100%), with CNN+LSTM performing well above chance (~80%). These results indicate that assessing whether an object is the ‘biggest’/’smallest’ in the scene is a relatively simple task even for basic models. This is encouraging since mastering superlatives requires several core skills that also underlie the use of positive GAs, namely (1) object identification, (2) object size estimation, and (3) object size ordering.

**POS₁/POS** CNN+LSTM+SA and FiLM obtain similar, very high max accuracy (~94%) in these two tasks, with CNN+LSTM being only slightly above chance level (~57%). Note that in this case we should not expect models to obtain 100% accuracy: while the min/max function in SUP is precise, the POS threshold function is vague (giving rise to borderline cases). The performance to be expected with sharp $k = 0.29$ is 97% for these two tasks (i.e., only in 3% of cases $k$ is assigned a value that makes the truth/falsity of a sentence different from what it would be with $k = 0.29$). The performance of the two top models is thus 3% below average ceiling accuracy (94% vs. 97%). These results, thus, indicate that CNN+LSTM+SA and FiLM are able to learn the threshold function that makes an object count as ‘big’/‘small’ in a given visual scene, while CNN+LSTM is not.⁶

**SET+POS** As before, CNN+LSTM performs only slightly above chance. Both FiLM and CNN+LSTM+SA experience a drop in performance compared to POS₁/POS, confirming that computing the threshold function over a subset of objects is the most challenging task. Applying the threshold function to the entire visual scene would yield ~65% accuracy in this dataset. This indicates that none of the two top models uses this strategy simplistically, as their performance is well above this result (max 81% for CNN+LSTM+SA and 92% for FiLM).⁷ FiLM neatly outperforms CNN+LSTM+SA (+11% in both max and average accuracy), thus showing a more robust pattern of results across tasks and a higher ability to handle complex reasoning problems.

---

<table>
<thead>
<tr>
<th>task</th>
<th>model</th>
<th>max val (pixels)</th>
<th>accuracy</th>
<th>hyper-parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CNN-LSTM</td>
<td>0.841</td>
<td>avg val ± sd</td>
<td>avg test ± sd</td>
</tr>
<tr>
<td>SUP₁</td>
<td>CNN-LSTM-SA</td>
<td>0.8153 ± 0.033</td>
<td>0.999 ± 0</td>
<td>0.9866 ± 0.033</td>
</tr>
<tr>
<td></td>
<td>FilM</td>
<td>0.9991 ± 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8673 ± 0.085</td>
<td>0.9306 ± 0.002</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8546 ± 0.086</td>
<td>0.9396 ± 0.001</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.941 (0.9565)</td>
<td>0.9545 ± 0.013</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.945 (0.9475)</td>
<td>0.9495 ± 0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>POS₁</td>
<td>CNN-LSTM-SA</td>
<td>0.5493 ± 0.009</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>FilM</td>
<td>0.5493 ± 0.007</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8673 ± 0.85</td>
<td>0.8546 ± 0.086</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.941 ± 0.002</td>
<td>0.94 ± 0.002</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.9375 ± 0.04</td>
<td>0.933 ± 0.002</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.945 (0.9475)</td>
<td>0.945 ± 0.9475</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.5455 ± 0.013</td>
<td>0.945 ± 0.002</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.9386 ± 0.026</td>
<td>0.9386 ± 0.026</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.9375 ± 0.04</td>
<td>0.933 ± 0.002</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.945 (0.9475)</td>
<td>0.945 ± 0.9475</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.945 ± 0.02</td>
<td>0.945 ± 0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.8485 ± 0.027</td>
<td>0.8788 ± 0.021</td>
<td></td>
<td></td>
</tr>
<tr>
<td>all</td>
<td>chance</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 1: Results by each model (column 2) in each task (1). Note that max val (3) refers to the highest accuracy obtained in the task by a given architecture across 192 runs (in brackets, highest accuracy obtained across 192 runs by FiLM trained from raw pixels), while avg val ± sd (4) and avg test ± sd (5) refer to the average accuracy (and relative standard deviation) obtained across 3 runs by the best configuration of hyper-parameters of a given architecture on, respectively, val and test set. As for the hyper-parameters (6), we report value of learning rate (lr), dropout (drop), and use/not use of batch normalization (b norm). Values in bold are the highest in the column.

⁶FiLM’s performance in POS₁ (where each scene contains only one shape type, but all shape types are seen across images) is rather unstable across runs (high SD). Since this is not so in POS, we conjecture that it may be due to the model learning shape-specific strategies in some runs of POS₁.

⁷As in POS₁/POS, ceiling performance with fixed $k = 0.29$ is 97% in this dataset.
Table 2: **Average accuracy** across 3 runs by a model in a hard test set. Models are trained on the task in train. Values in bold are the highest in the column.

<table>
<thead>
<tr>
<th>hard test (train)</th>
<th>model</th>
<th>avg acc ± sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>POS-hard (POS)</td>
<td>CNN+LSTM</td>
<td>0.5375 ± 0.01</td>
</tr>
<tr>
<td></td>
<td>CNN+LSTM+SA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FiLM</td>
<td></td>
</tr>
<tr>
<td>SET+POS-hard (SET+POS)</td>
<td>CNN+LSTM</td>
<td>0.4623 ± 0.004</td>
</tr>
<tr>
<td></td>
<td>CNN+LSTM+SA</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FiLM</td>
<td></td>
</tr>
</tbody>
</table>

| all               | chance              | 0.5          |

While these results are very encouraging, they might be the outcome of exploiting biases in the data rather than learning the semantic components that make up the meaning of GAs according to linguistic theory and psycholinguistic evidence. For example, identifying the reference subset but applying the max/min function (rather than a threshold function) to this set would yield a remarkable 92% accuracy, which is on a par with the top result by FiLM. In what follows, we investigate the abilities of the trained models in more depth by testing them on a number of diagnostic test sets.

7 **Analysis**

In this section, we aim at better understanding what the models learn (or do not learn) when performing the different tasks. To do so, we carry out a stress-test and a compositionality analysis.

7.1 **Stress-Test: Hard Contexts**

To test to what extent models master the core abilities that are arguably needed to perform POS and SET+POS, we build two hard diagnostic test sets which make the use of other strategies not effective. Similarly to the main datasets, these hard test sets include 2K perfectly-balanced datapoints.

**POS-hard** We explore whether models trained on POS properly learn to compute the threshold function rather than using a simplistic strategy such as applying the min/max function over the visual context. Indeed, it has been proposed that young children might use positive-form GAs as superlatives in early stages of language acquisition (Clark, 1970). To test whether this explains (part of) the results on POS, we build a hard test set identical to POS except for the fact that objects that are either the biggest or the smallest in the entire visual scene are never queried. Thus, in POS-hard systematically using the superlative strategy would result in chance level scores. As reported in Table 2, the accuracy obtained by CNN+LSTM+SA and FiLM does not dramatically decrease compared to POS (~94% vs. ~87%). The drop can be explained by the fact that in POS-hard we query objects that are overall closer to the threshold, thus increasing the number of more difficult, borderline cases. This test thus confirms that CNN+LSTM+SA and FiLM do learn to generally compute and apply the threshold function over an entire visual scene.

**SET+POS-hard** To investigate whether models trained on SET+POS learn to identify the reference subset and to apply the threshold function to it, we build a hard test set identical to SET+POS except for the fact that objects that are the biggest or the smallest in the reference set are never queried. Thus, in SET+POS-hard applying the min/max function to the reference set would lead to chance level. As shown in Table 2, only FiLM is above chance level in this test set, with 65% accuracy. Both CNN+LSTM and CNN+LSTM+SA obtain scores slightly below chance (~30% for CNN+LSTM+SA compared to SET+POS). This is a striking result, which reveals that the accuracy scores achieved by CNN+LSTM+SA in SET+POS must be due to shortcut strategies.

While we do not have full understanding of what is being exploited by the CNN+LSTM+SA model, we do observe a bias towards predicting that an object counts as ‘small’. As shown in Figure 4, the model obtains high accuracy on sentences involving small objects (small true, big

---

8 This also leads to slightly lower average ceiling accuracy in POS-hard: performance with fixed k = 0.29 is now ~92% in contrast to ~97% in POS.
false, while its accuracy on sentences targeting big objects (big true, small false) is below chance level. Indeed, the model predicts false for big objects and true for small objects in 73% of cases.

As for FiLM, its pattern of accuracy is overall more stable across sentence types, though big objects (big true, small false) are still significantly harder than small ones (small true, big false). Bigger objects are more challenging because by definition they are closer to the threshold (as can clearly be seen in Figure 1), which in turn means that they are more likely to be borderline cases. The plots in Figure 5 illustrate this effect, showing that objects that count as ‘big’ are more likely to flip the truth value of a sentence due to the fuzziness of the threshold. FiLM’s results are thus to be expected, and in fact encouraging: if a model is correctly learning to apply the threshold function to the reference set, most of the errors should involve borderline objects. This is confirmed: see Figure 6, where correct and wrong predictions by best run of FiLM (73% acc.) against distance from the threshold. Best viewed in color.

Finally, the evidence that models obtain slightly different results across shapes (see Figure 4) could suggest that models learn different, shape-specific representations of ‘big’ and ‘small’. The following analysis precisely aims at exploring this issue.

7.2 Compositionality: Unseen Combinations

In our last analysis, we investigate whether the models learn an abstract representation of GAs that can be compositionally applied to unseen adjective-noun combinations. We focus on the SET+POS task and extract a subset of the corresponding dataset in which each size adjective appears only with two nouns denoting two distinct shape types: ‘big (circle|rectangle)’, ‘small (square|triangle)’. This subset of the data amounts to half of SET+POS and thus contains 10K datapoints, that we split into train (8K), val (1K), and test (1K). We refer to this test set as seen (its adjective-noun combinations are seen in training). We then create a second, unseen test set with 1K datapoints, where the adjectives appear with different nouns: ‘big (square|triangle)’, ‘small (circle|rectangle)’. We train all models three times using their best hyper-parameter configurations and evaluate them on both seen and unseen. If
models learn an abstract representation of GAs that includes a variable for the noun they modify, we should observe no difference in performance between the two test sets.

As reported in Table 3, this turns out not to be the case. While in the seen test set all models obtain similar accuracies to those obtained in SET+POS, in unseen their performance is not only well below chance level, but it is in fact the inverse of their results on seen: all instances which are predicted correctly in seen are incorrectly predicted in unseen (e.g., FiLM obtains $\sim85\%$ on seen and $\sim15\%$ on unseen). This suggests that the models learn a default strategy per noun rather than an abstract adjective representation that generalizes to unseen combinations. For example, the models predict true any time the size of a circle or a rectangle exceeds a certain threshold, regardless of whether the noun is modified by ‘big’ (seen combinations) or ‘small’ (unseen combinations).

Table 3: Compositional task. Results by the models.

<table>
<thead>
<tr>
<th>model</th>
<th>avg seen ± sd</th>
<th>avg unseen ± sd</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN+LSTM</td>
<td>0.608 ± 0.01</td>
<td>0.4036 ± 0.003</td>
</tr>
<tr>
<td>CNN+LSTM+SA</td>
<td>0.7813 ± 0.009</td>
<td>0.235 ± 0.006</td>
</tr>
<tr>
<td>FiLM</td>
<td>0.8489 ± 0.014</td>
<td>0.153 ± 0.02</td>
</tr>
<tr>
<td>chance</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

8 Discussion

We tackle the modeling of size GAs as a relational problem in the domain of visual reasoning, and show (in contrast with Kuhnle et al., 2018) that FiLM is able to learn the function underlying the meaning of these expressions. However, none of the models develop an abstract representation of GAs that can be applied compositionally, an ability that even 4-year-olds master (Barner and Snedeker, 2008). This is in line with recent evidence showing that deep learning models do not rely on systematic compositional rules (Baroni, 2019; Bahdanau et al., 2019).

An interesting open question, which we plan to explore in future work, is whether training models to jointly learn superlative, comparative, and positive GAs (similarly to how Pezzelle et al. (2018) did for quantities), or framing the task in a dialogue setting (as Monroe et al. (2017) did for colors) could lead to more compositional models. Moreover, it might be worth exploring whether equipping models with similar inductive biases as those leading speakers of any language to develop abstract, compositional representations of size adjectives is needed to properly handle these expressions. In parallel, the present work could be extended to other GAs and threshold functions.

In the long term, we aim to move to natural images. This requires world knowledge, a confounding factor intentionally abstracted away in synthetic data. Since children learn to exploit world knowledge after mastering the perceptual context (Tribushinina, 2013), adopting an incremental approach might be promising.

Acknowledgments

We kindly thank Raffaella Bernardi and Elia Bruni for insightful discussions and their comments on an early draft of this paper. Alexander Kuhnle for sharing opinions and details regarding the models, and the audience at talks given at the ILLC and at the Blackbox@NL Workshop held at the Jheronimus Academy of Data Science. We are very grateful to the anonymous reviewers of EMNLP-IJCNLP for their valuable feedback. This work was funded by the Netherlands Organisation for Scientific Research (NWO) under VIDI grant no. 276-89-008, Asymmetry in Conversation.
References


