Probing Cross-Modal Representations in Multi-Step Relational Reasoning

Iuliia Parfenova[*], **Desmond Elliott**[†], **Raquel Fernández**[‡], **Sandro Pezzelle**[‡]
[*]Department of Computer Science, Vrije Universiteit Amsterdam
[†]Department of Computer Science, University of Copenhagen
[‡]Institute for Logic, Language and Computation, University of Amsterdam
research@julia.jig-san.me, de@di.ku.dk,
{raquel.fernandez|s.pezzelle}@uva.nl

Abstract

We investigate the representations learned by vision and language models in tasks that require relational reasoning. Focusing on the problem of assessing the relative size of objects in abstract visual contexts, we analyse both one-step and two-step reasoning. For the latter, we construct a new dataset of three-image scenes and define a task that requires reasoning at the level of the individual images and across images in a scene. We probe the learned model representations using diagnostic classifiers. Our experiments show that pretrained multimodal transformer-based architectures can perform higher-level relational reasoning, and are able to learn representations for novel tasks and data that are very different from what was seen in pretraining.

1 Introduction

Intelligence is classically described as "*the ability to see the similarities among dissimilar things and the dissimilarities among similar things*" (Thomas Acquinas, 1225-1274, reported by Ruiz, 2011). Developing systems that can reason over objects and their relations is indeed a long-standing goal of artificial intelligence research, as argued by Johnson et al. (2017). In recent years, huge progress toward this goal has been made in the language and vision community. Starting from Malinowski and Fritz (2014) and Antol et al. (2015), a wealth of studies have focused on language-driven visual reasoning, namely the problem of reasoning about an image given some linguistic input.

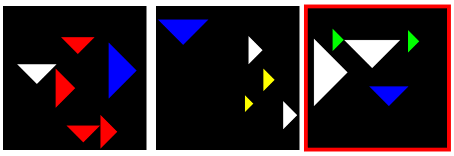
Generally speaking, there are two main types of problems in visual reasoning datasets (see Santoro et al., 2017): *non-relational*, requiring models to focus only on a given object (e.g., answering the question "*What material is the cube made of?*"), and *relational*, requiring models to pay attention to several or even all the objects in the image (e.g., indicating whether the statement "*There are four cubes that are red*" is true or false). Relational problems call for higher-level abilities, such as counting or directly comparing objects, both of which involve recognising the (dis)similarities among things.

In this paper, we focus on an important but understudied, relational reasoning task: assessing the relative size of objects in visual contexts, that is, determining whether an object counts as 'big' or 'small' in an image. We define a *multi-step* relational reasoning problem formulated as a sentence verification task. We construct a dataset of three-image scenes where a given target object, e.g., a blue triangle, is present in each image: two images have target objects with the same contextually-defined size and one image stands out in this regard. The task requires verifying whether a simple natural language statement standing for a first-order logical form describes a scene, e.g., "*There is exactly one blue triangle that is small in its image in this scene*" (Figure 1). Such multi-step relational reasoning is at play in many real-life situations: e.g., the same exact pan may count as 'big' in all contexts except a restaurant kitchen.

We experiment with two types of models to solve this task: a modular neural network (Hu et al., 2017) and LXMERT, a pre-trained multimodal transformer (Tan and Bansal, 2019). We probe the learned representations of LXMERT to assess whether, and to what extent, it has learned the underlying structure of the data. By means of two experiments with probing classifiers (Alain and Bengio, 2017; Hupkes et al., 2018; Belinkov and Glass, 2019), we first verify that it is able to perform the task at the image level (i.e., to compute the relative size of the target object at the image level); then, we test its ability to reason at the multi-image level and detect the image that stands out.

The experiments show that LXMERT is able to solve the multi-step relational reasoning task

Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021), pages 152–162
Bangkok, Thailand (Online), August 6, 2021. ©2021 Association for Computational Linguistics



there is exactly one blue triangle that is small in its image in this scene *T*
there is exactly one blue triangle that is big in its image in this scene *F*
there are exactly two blue triangles that are small in their images in this scene *F*
there are exactly two blue triangles that are big in their images in this scene *T*

Figure 1: One sample scene from our dataset and the four statements it can be paired with, including corresponding truth values assigned as explained in Section 4.1. For clarity, the odd-one-out image (holding the *odd* size) is framed in red. Best viewed in color.

with an accuracy of 88.8%, and that the majority of errors occur when the relative size of the target object is difficult to determine. Our analyses show that (i) in most cases, different attention heads in LXMERT specialise to localising the smallest and biggest objects in the images, (ii) that the cross-modal representations learned appear encode a threshold function that controls whether an object is 'big' or 'small' in an image, and (iii) that a simple diagnostic classifier successfully identifies the instance that stands out in a three-image scene. Taken together, these findings lend further support to the advanced reasoning abilities of pre-trained transformer-based architectures, showing that they can perform higher-level relational reasoning and are able to deal with novel tasks and novel data, including synthetic data not available during pre-training.[1]

2 Problem Formulation

We investigate multi-step relational reasoning by formulating the problem as a visually grounded sentence verification task (see Figure 1). Given a pair $\langle \texttt{scene}, \texttt{statement} \rangle$ consisting of a visual scene and a statement about such scene, the task consists in classifying the statement as either `true` or `false`. In our setup, a `scene` consists of 3 images: $\langle \texttt{img}_1, \texttt{img}_2, \texttt{img}_3 \rangle$, each including an instance of the target object (e.g., a blue triangle) together with other geometrical shapes of the same type (e.g., triangles of other colours). A `statement` paired with a scene is of the following form: "*there is exactly one blue triangle that is small in its image in this scene*" or "*there are exactly two blue triangles that are big in their images in this scene*". As we will explain in detail in Sec. 4.1, the dataset is created such that the target object counts as either 'big' or 'small' *in only one* of the three images in a scene.

Arguably, solving the task requires the following two steps of relational reasoning: (1) identifying whether the target object counts as either 'big' or 'small' in each image, and (2) counting how many images include a big/small target. However, in our setup there is no direct supervision for any of these steps. In other words, the training data does not indicate which images contain an object that counts as big/small nor explicitly how many images contain a big/small target.

3 Related Work

3.1 Visual Reasoning

To evaluate *reasoning* abilities of multimodal models, several datasets of synthetic scenes and questions, such as CLEVR (Johnson et al., 2017), ShapeWorld (Kuhnle and Copestake, 2017), and MALeViC (Pezzelle and Fernández, 2019) have been proposed in recent years. Our work directly builds on them, and particularly on approaches adopting a multi-image setting, such as NLVR (Suhr et al., 2017) and NLVR2 (which, however, contains pairs of natural scenes; Suhr et al., 2019). In NLVR, in particular, a crowdsourced statement is coupled with a synthetic scene including 3 independent images, and models must verify whether the statement is true or false with respect to the entire visual input. This involves handling phenomena such as counting, negation or comparisons, that require perform *relational* reasoning over the entire scene, e.g.: *There is a black item in every box*, *There is a tower with yellow base*, etc. However, most $\langle scene, statement \rangle$ pairs do not challenge models to do the same at the level of the single image (or *box*), where a low-level understanding of the object(s) of interest (shape, color, etc.) often suffices. Our approach is novel since it requires two steps of *relational* reasoning: at the level of both the single image and the multi-image context.

3.2 Multi-Image Approaches

Our approach is also related to other work in language and vision involving multiple images. One is the *spot-the-difference* task: in Jhamtani and Berg-Kirkpatrick (2018), models are fed with pairs of video-surveillance images that only differ in one detail, and asked to generate text which de-

[1] The code to generate the data, and to train and evaluate the models, is available at https://github.com/jig-san/multi-step-size-reasoning.

scribes such difference. The same task—with different real-scene datasets—is explored by Forbes et al. (2019) and Su et al. (2017); others experiment with pairs of similar images drawn from CLEVR (Johnson et al., 2017) or similar synthetic 3D datasets (Park et al., 2019; Qiu et al., 2020). This task is akin to ours since it requires a higher-level reasoning step: systems must *reason* over the two independent representations to describe what is different. However, in practice, it does not always require semantic understanding (Jhamtani and Berg-Kirkpatrick, 2018); when it does, the changes often involve one object's *fixed* attribute (color, shape, material, etc.) rather than a *contextually-defined* property whose applicability depends on the other objects in the image.[2]

A similar, partially overlapping task is *discriminative captioning*: systems are fed with a set of similar images and asked to provide a description that unequivocally refers to a target one. Many approaches have been proposed focusing on synthetic (Andreas and Klein, 2016; Achlioptas et al., 2019) or natural scenes (Vedantam et al., 2017; Cohn-Gordon et al., 2018; Vered et al., 2019), very often embedding pragmatic components based on the Rational Speech Acts framework (RSA; Goodman and Frank, 2016). Also in this case, however, differences among images mainly involve *intrinsic* attributes of the objects rather than relational properties defined at the level of the image.

4 Method

4.1 3POS1 Dataset

Our dataset is based on the POS1 dataset from MALeViC (Pezzelle and Fernández, 2019), in which images contain 4 to 9 same-shape objects, e.g., squares. Each object is labeled with a ground-truth relative size, indicating whether the object counts as either *big* or *small* in that particular context. The label is determined by the following threshold function motivated by cognitive science studies on how humans interpret relative gradable adjectives (Schmidt et al., 2009):

$$T = \text{Max} - k(\text{Max} - \text{Min}) \qquad (1)$$

where Max and Min represent the areas of the biggest and smallest objects in the image, and k is a positive value < 0.5.[3] Thus, an object with a certain area can count as *big* in one image and as *small* in another one. In total, the POS1 dataset contains 20K $\langle image, statement\rangle$ datapoints (16K train, 2K val, 2K test), where statements are about the size of a *target* object based on its unique color: e.g., "*the blue triangle is a small triangle*".

The dataset for the present experiments, which we name 3POS1, is constructed as follows: For each image in each split of POS1, we randomly sample two images from that split with the same target object (e.g., a blue triangle) but the opposite ground-truth size (e.g., *big*). We obtain 20K sets of three images where one size is *prevalent*, i.e., present in two images, and one is *odd*, i.e., held by only one image.[4] The sizes *big* and *small* are the prevalent ones in 10K cases each, thus the dataset is balanced. Then, for each three-image `scene`, we generate four logic-based templated statements, two of which are *true* and two *false* for the given `scene`.[5] The only variation in the statements is the target object. The four types of statement are (alongside examples with respect to Figure 1):

(i) one $\langle shape, color\rangle$ small:
"*There is exactly one blue triangle that is small in its image in this scene*" $\rightarrow$ `True`

(ii) one $\langle shape, color\rangle$ big:
"*There is exactly one blue triangle that is big in its image in this scene*" $\rightarrow$ `False`

(iii) two $\langle shapes, color\rangle$ small:
"*There are exactly two blue triangles that are small in their images in this scene*" $\rightarrow$ `False`

(iv) two $\langle shapes, color\rangle$ big:
"*There are exactly two blue triangles that are big in their images in this scene*" $\rightarrow$ `True`

4.2 Models

To tackle the visually grounded sentence verification task, we use two models that achieve state of the art results on the NLVR (Suhr et al., 2017) and NLVR2 (Suhr et al., 2019) tasks, respectively: N2NMN (Hu et al., 2017) and LXMERT (Tan and Bansal, 2019). The End-to-End Module Network

[2] One notable exception is position (Park et al., 2019; Qiu et al., 2020), which can involve spatial relations of objects.

[3] To account for gradable adjectives' *vagueness*, for each image k was randomly sampled from the normal distribution centered on 0.29, the best-predictive value in Schmidt et al. (2009). See Pezzelle and Fernández (2019) for further details.

[4] On average, each target image appears 2 times as a distractor in the dataset (min: 0, max: 10). The position of the odd-one-out image in the scene is assigned randomly.

[5] The odd-one-out is the same for all statements; see Fig. 1.

(N2NMN), belongs to the family of *modular* networks, which treat a sentence as a collection of predefined subproblems (e.g., counting, localization, conjunction, etc.), each handled by a dedicated *module*. Compared to its direct predecessor NMN (Andreas et al., 2016), in particular, N2NMN does not require any external supervision (e.g., a parser) to process the sentence into its components. The latter, Learning Cross-Modality Encoder Representations from Transformers (LXMERT), is a transformer-based multimodal architecture pre-trained on several language-and-vision tasks; as such, it is claimed to be *universal*, that is, capable of solving virtually any visual reasoning problem. LXMERT uses BERT (Devlin et al., 2019) to encode the language input; as for the image, it considers the sequence of N salient regions output by Faster R-CNN (Ren et al., 2015).

To assess the suitability of these models for the 3POS1 task, we first evaluate them on the original POS1 task where statements are evaluated against a single image. For N2NMN, we use a public implementation,[6] specifically, the code developed for training and an evaluating the model on the CLEVR dataset (Johnson et al., 2017). For LXMERT, we use a snapshot pre-trained on several multi-modal tasks,[7] that we fine-tune using the training set of POS1. The ceiling performance for this task is 97% accuracy (using a fixed interpretation of the threshold parameter $k = 0.29$). LXMERT achieves 93.4% accuracy, which outperforms both N2NMN (78.1%) and the models tested by Pezzelle and Fernández (2019). This shows the overall advantage of transformer-based architectures over competing methods, in line with previous findings (Devlin et al., 2019). Moreover, it indicates the capability of LXMERT—which is pre-trained on natural images and language—to deal with synthetic data after fine-tuning (crucially, when not fine-tuned it yields an accuracy of 50%, i.e., random). Based on its performance, we focus on LXMERT in the main experiments and analyses in this paper.

4.3 Experimental Setup

We fine-tune LXMERT on the 3POS1 dataset by adapting the method applied by Suhr et al. (2019) for the two-image scenes of NLVR2 to our three-image scenes. More concretely, each datum in 3POS1 is composed of 3 images $\langle \texttt{img}_0, \texttt{img}_1, \texttt{img}_2 \rangle$, a statement $\texttt{stat}$, and a ground truth label $\texttt{True}$ or $\texttt{False}$. Recall, that the visually grounded sentence verification task is to predict a label ($\texttt{True}$ or $\texttt{False}$), given a representation of the images and the statement. An overview of how this is achieved with LXMERT is shown in Figure 2. First, visual features are extracted separately for each image with Faster R-CNN (Ren et al., 2015). Then cross-modal representations $\mathbf{x_i}$ are extracted from the [CLS] from the LXMERT encoder for each image in a scene:

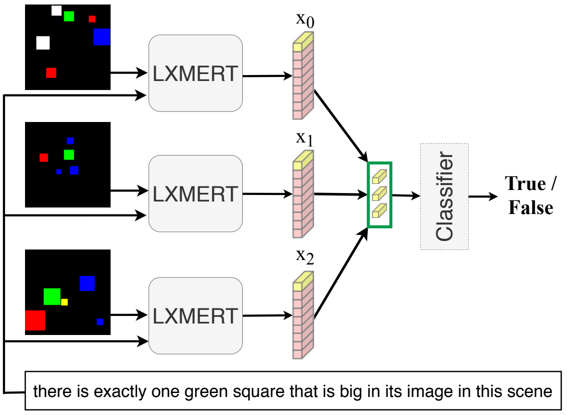

LXMERT
LXMERT
LXMERT
x0
x1
x2
Classifier
True / False
there is exactly one green square that is big in its image in this scene

Figure 2: Overview of our visually-grounded sentence verification model. Given a three-image scene and a statement, LXMERT encodes each image–statement pair separately, from which a single cross-modal representation is extracted from the special [CLS] token (shown in yellow). These [CLS] representations are concatenated and propagated through a non-linear classifier to predict whether the statement accurately describes the scene.

$$\begin{aligned} \mathbf{x_0} &= \text{lxmert\_encoder}(\texttt{img}_0, \texttt{stat}) \\ \mathbf{x_1} &= \text{lxmert\_encoder}(\texttt{img}_1, \texttt{stat}) \\ \mathbf{x_2} &= \text{lxmert\_encoder}(\texttt{img}_2, \texttt{stat}) \end{aligned} \tag{2}$$

For label prediction, we train a classifier on the concatenation of the three image–statement representations (Eqn. 3), followed by a linear layer with learned parameters $\mathbf{W}$ and a bias vector $\mathbf{b}$ (Eqn. 4), followed by layer normalization (Ba et al., 2016) and a GeLU activation (Hendrycks and Gimpel, 2016) (Eqn. 5), and finally, a sigmoid activation function over a linear layer with learned parameters

[6] https://github.com/ronghanghu/n2nmn.

[7] Downloaded from http://nlp1.cs.unc.edu/data/model_LXRT.pth

	test accuracy	
statement type	*true*	*false*
one $\langle shape, color\rangle$ big	0.868	0.876
two $\langle shapes, color\rangle$ big	0.880	0.908
one $\langle shape, color\rangle$ small	0.872	0.900
two $\langle shapes, color\rangle$ small	0.876	0.924
overall	0.888	

Table 1: LXMERT results on the test set of 3POS1 by the best model's run, split by statement type.

$\mathbf{W_1}$ and a bias vector $\mathbf{b_1}$ (Eqn. 6):[8]

$$\mathbf{c} = [\mathbf{x_0}; \mathbf{x_1}; \mathbf{x_2}] \tag{3}$$

$$\mathbf{z} = \mathbf{Wc} + \mathbf{b} \tag{4}$$

$$\mathbf{z_1} = \text{LayerNorm}(\text{GeLU}(\mathbf{z})) \tag{5}$$

$$\mathbf{y} = \sigma(\mathbf{W_1 z_1} + \mathbf{b_1}) \tag{6}$$

The LXMERT encoder and the classifier are fine-tuned for 12 epochs to prevent overfitting with a batch size 64. The learning rate of the Adam optimizer (Kingma and Ba, 2014) is 5e-5. The fine-tuning is performed for 5 random seeds.

5 Results

Overall, LXMERT achieves a very high accuracy on the task, averaged across 5 runs: 0.8909 ± 0.004 in validation set, 0.8864 ± 0.005 in test set. Moreover, its performance turns out to be fairly stable across various statement types, with the best model run's accuracy (see Table 1) ranging from 0.868 (one $\langle shape, color\rangle$ big, *true*) to 0.924 (two $\langle shapes, color\rangle$ small, $false$). Interestingly, for all four statement types, the model experiences a slight advantage with *false* over *true* statements, even though the dataset was carefully balanced. Taken together, these results indicate that the model, which is pre-trained on natural images, can deal with the synthetic scenes in our dataset after fine-tuning. This is in line with the claim that off-the-shelf transformer-based models can be applied to a wide range of different learning problems and data. At the same time, the model yields random accuracy when not fine-tuned, which reveals that our new dataset is challenging and involves a type of reasoning not captured during pre-training.

In Pezzelle and Fernández (2019), models were shown to make more errors when the area of the queried object is closer to the threshold (see Eq. 1). We check if this is the case also for LXMERT on our 3POS1 task. To do so, we consider the cases where the model gives a wrong prediction. Among the 3 images in a scene, we take the one with the lowest distance from the threshold. We then check whether the model makes more errors when such distance is lower, i.e., when there is at least one image in the scene with a *borderline* size. As reported in Table 2, this is indeed the case: 75% of incorrect predictions involve cases where (at least) in one image the target object is close to the threshold (< 0.1) (see Figure 3 , where the leftmost image is *borderline*). In contrast, only around 3% of the errors involve clear-cut cases, i.e., images where the target object's distance from threshold is ≥ 0.2. As observed by Pezzelle and Fernández (2019), this may suggest that the model is genuinely learning to compute the threshold function based on the areas of the relevant objects in the scene. Further support for this is given by the performance of the model on the 15 cases in the test set where the target object has the same area in the three-image scene. These cases could be expected to act as a confound for the model,[9] but LXMERT succeeds in 14/15 cases. Consistently with the error pattern reported above, the missed case contains low-distance objects (the lowest distance is equal to 0.1). In the next section, we more extensively explore this issue.

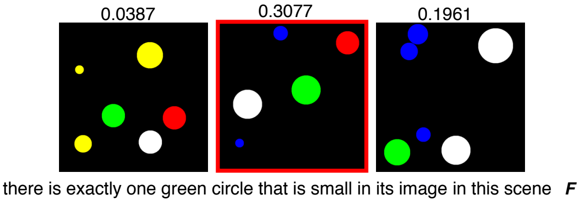

0.0387
0.3077
0.1961
there is exactly one green circle that is small in its image in this scene *F*

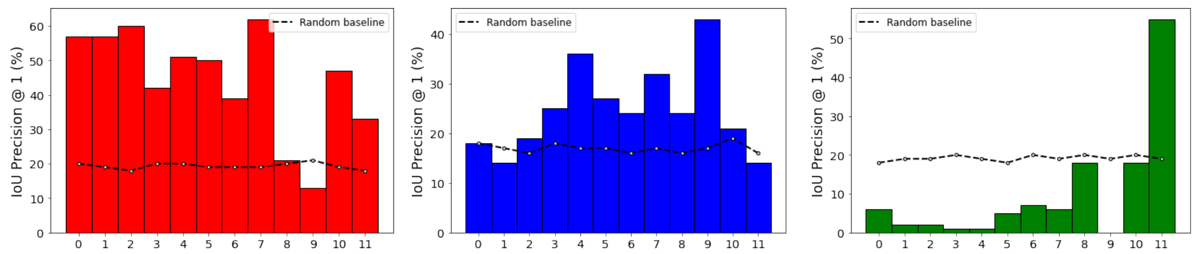
Figure 3: A sample from the test split of 3POS1, for which LXMERT predicts the incorrect label (*True*, instead of *False*). The numbers above the images are the distances of the target object (*green circle*) from the image-specific threshold. Here, the target object in the leftmost image is very close to that image's threshold value, so it is challenging for the model to detect whether it is *big* or *small*. The odd-one-out image is framed in red. Best viewed in color.

6 Analysis at the Individual Image Level

Our results show that LXMERT achieves a high level of accuracy on our visually-grounded sentence verification task on the three-image 3POS1

[8] This is identical to the approach followed by Tan and Bansal (2019) to finetune LXMERT for NLVR2 classification.

[9] The target objects have exactly the same area in pixels but each target object has its own context-defined size.



(a) Target object

(b) Smallest object

(c) Biggest object

Figure 4: Intersection over Union Precision at K=1, per attention head (in the x-axis), for the target object in an image (a), the smallest object in an image (b), and the largest object in an image (c).

threshold distance	< 0.1	< 0.15	< 0.2	≥ 0.2	total
% of errors	75.89	13.84	7.14	3.13	100
number of cases	170	31	16	7	224

Table 2: Analysis of LXMERT's errors with respect to target object's distance from the threshold. Threshold distance refers to the lowest value in the visual scene.

dataset. In this section, we investigate how the model may be solving the task. Specifically, we explore what visual information the model attends to within each image and whether the representations learned by the model encode information about the context-dependent threshold that determines what counts as *big* or *small* in a given image.

6.1 Visual Attention over Key Object Types

Recall that the ground truth labels in our dataset are assigned based on the function in Eqn. 1, which was shown to fit well with human judgements about relative gradable adjectives (Schmidt et al., 2009). This function computes a threshold value taking into account the biggest and smallest objects in the context of an image. Thus, a possible strategy adopted by the model at the level of individual images could be to identify the target object and reason about the context by focusing on the biggest and smallest objects. We test this hypothesis by checking whether the model pays particular attention to these object types (target, biggest, smallest) or whether its attention is rather uniformly distributed over all regions detected by Faster R-CNN (Ren et al., 2015).

To compute which objects are the most attended, we use the Intersection over Union (IoU) metric (Russakovsky et al., 2015). We take the attention weights provided by the `[CLS]` token representation, extracted from the final layer of the best fine-tuned model with frozen parameters. We then use IoU Precision @ K to find the percentage of the labels correctly predicted by the model using the following steps:

1. **Extract top-K object proposals:** For each correctly predicted label, separately for each of the three images in a scene, we take the object proposals of the image regions detected by Faster R-CNN with K-highest scores in the `[CLS]` token. We perform the procedure for each attention head of the representation, extracted from the cross-modality encoder for the corresponding visual-language input. We ignore the object proposals related to the background areas of the image, which we identify based on the labels provided by Faster R-CNN.[10]
2. **Extract ground-truth bounding boxes:** We take the ground-truth bounding boxes of the biggest/the smallest/target objects from all three images in the input scene.[11]
3. **Calculate Pairwise IoU:** We calculate the pairwise IoU between the top-K object proposals and the ground truth bounding boxes, obtained in Steps 1 and 2. We take the highest IoU value calculated for all these pairs.
4. **Calculate IoU Precision@K:** The IoU precision @ K is the percentage of all the IoU values obtained in Step 3 that are > 0.5.

We also compute a random baseline for all three categories with the same steps, except in Step 1 we randomly select K objects from the 36 detected by Faster R-CNN, instead of using the ones with the highest attention scores.

We use the smallest possible value for K = 1, as the most illustrative case in which the metric only

[10] The attributes predicted for the regions corresponding to the black background in our scenes could be *black* or *dark*.

[11] We calculate the coordinates of the boxes using objects position and radius provided in the annotation of the POS1 dataset by Pezzelle and Fernández (2019).

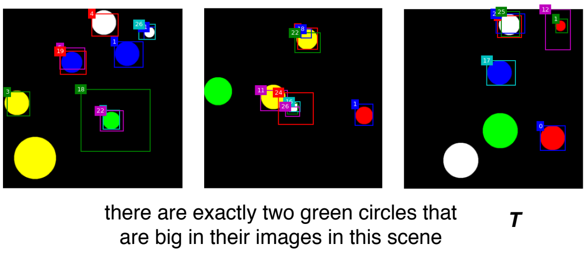

there are exactly two green circles that
are big in their images in this scene
T

Figure 5: Example of object proposals most attended to by the 9th head of the last layer of the cross-modality encoder. In each image, the model attends to all of the objects except the biggest ones. Simultaneously, in the leftmost image, it also focuses on the *green circle*, which is the target object in this scene.

looks at the single object in each image to which the model attends the most.

Figure 4 shows the results of the IoU Precision @ K for the 12 attention heads in LXMERT. In particular, Figure 4a shows that many of the attention heads attend to the target object that is queried directly in the input sentence. Figures 4b and 4c demonstrate that the model also looks at the surrounding visual context, which is needed to perform relational reasoning. A comparison of behaviour across the Figures reveals that different attention heads *appear* to specialise on different object types: attention head 9 learns to attend to the smallest objects while it pays no attention to the biggest objects and less than random attention to the target objects. We also highlight the observed behaviour of attention head 11, which is the only head that reliably attends to the biggest objects.

Figure 5 shows an example of the objects attended to by attention head 9 in one sample scene. Here, we can see that the model is primarily attending to the smallest objects in the scene.

6.2 Implicit Knowledge of the Threshold

The analysis above showed that the model, besides the target object, also pays attention to key contextual information, particularly to the smallest and biggest objects in an image. These objects are critical to compute the threshold to determine if a target object is *big* or *small* relative to the context of an image. To test whether the representations learned by the model implicitly encode information about the context-dependent threshold, we use a diagnostic classifier (Alain and Bengio, 2017; Hupkes et al., 2018; Belinkov and Glass, 2019). Probing or diagnostic tests are useful tools to better understand the inner workings of deep models. Given a hypothesis about information that may be encoded by a trained model, a probe checks whether such information is accessible by a relatively simple classifier.

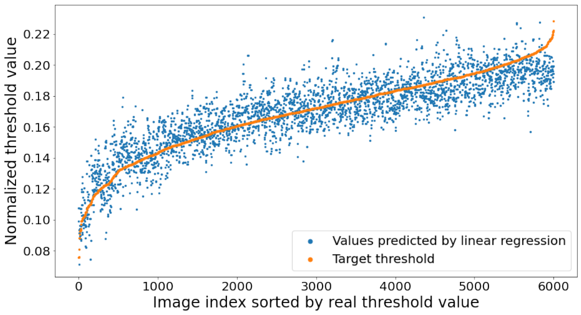


Figure 6: Comparison of threshold values predicted by the linear regression model (blue dots) with the actual threshold for each of the 6000 test images (orange dots). Here, the real target values are sorted in ascending order, and the predicted values are sorted with respect to the corresponding targets' indices. The thresholds are normalized by the area of the one image, with the square root transformation. Best viewed in color.

Concretely, in this experiment we use a linear regression classifier[12] to predict the threshold values for each of the three images in a scene given the cross-modality features learned by the LXMERT encoder (x_0, x_1, x_2 in Eqn. 2). The classifier uses the same train/val/test splits of the 3POS1 dataset. The predicted and actual values are displayed in Figure 6, which shows that a simple linear classifier can predict the threshold values for each image in a scene remarkably accurately (mean squared error on the test set is $6.64e-05$). This confirms that the cross-modality representations learned by the model are representing the threshold in each image.

7 Analysis at the Multi-Image Level

In the previous section, we analysed the model representations at the level of the independent images. Here, we probe the representations with respect to the entire three-image scene. First, we investigate whether the representations encode information on the overall configuration of the scene (Sec. 7.1). Second, we probe their effectiveness in identifying the odd-one-out image in the scene (Sec. 7.2). In both analyses, we use diagnostic classifiers,[13] that take as input the concatenation of the three image-statement cross-modal representations (Eqn. 3).

[12] Least squares linear regression from the `sklearn`.

[13] Trained on the same splits as the main experiments.

		sentence verification (LXMERT full model)	
		✓	✗
scene configuration classification (linear diagnostic)	✓	85.70	2.45
	✗	3.10	8.75

Table 3: Confusion matrix with % of scenes in the test set that are (in)correctly classified by the full LXMERT model for the original sentence verification task and by the linear SVM for the scene configuration task.

7.1 Scene Configuration Classification

We first investigate whether the representations learned by the model encode the *configuration* of the scene, that is, whether they are effective to distinguish between scenes where 1 target object counts as small and 2 as big (hence, *1small2big*), and vice versa (*1big2small*). In principle, this counting step is necessary to solve the sentence-verification task (see Sec. 2), and this probe determines whether the model is reasoning at the level of the scene or exploiting other strategies, such as capturing random correlations in the data.

We use an SVM classifier with linear kernel (Boser et al., 1992)[14] to probe the representations learned by the model, and find that they are indeed useful for predicting the configurations. Accuracy on the test set is 88.15%, which is well above chance level (50%). As reported in Table 3, in the large majority of cases (85.7%) a correct prediction in the sentence verification task corresponds to a correct assessment by the diagnostic classifier. This confirms that LXMERT learns representations that encode the configuration of the scene.

7.2 Odd-One-Out Image Identification

Our results so far show that the model is able to perform the multi-step sentence verification task with high accuracy and that the representations encode information about different configurations of scenes. However, there is yet no guarantee that the model is able to identify the odd-one-out image (i.e., the image that is not prevalent; see Sec. 4.1). We test this by means of another diagnostic classifier: given a scene representation, the task is to predict the position of the odd-one-out image (hence, OOO), namely image 0, 1, or 2.

We initially experiment with the same type of diagnostic classifier used in the previous analysis: an SVM with a linear kernel. However, this linear classifier was only able to accurately classify the position of odd-one-out images associated with `image–scene` instances labelled `True`, suggesting that the prediction of the position of the odd-one-out cannot be solved by a linear classifier. Therefore, we use a non-linear MLP and also report the results of a control task, where the labels are randomly assigned to the instances (Hewitt and Liang, 2019). The MLP is a two-layer neural network with 128 units in each layer followed by a ReLU activation function, and finally a learned projection into 3 output units, followed by a softmax normalisation. We train the MLP with a cross-entropy objective function for four epochs using the Adam optimiser with the default learning rate.

	train	**valid**	**test**
OOO	0.8767	0.8771	0.8659
control	0.3385	0.3386	0.3359

Table 4: Accuracy of the MLP diagnostic classifier on the train/val/test splits of the data on both the OOO and the control setting. Chance level is 0.33 for all splits.

Table 4 reports the results of the non-linear diagnostic classifier in both the OOO and *control* settings. As can be seen, while the MLP does not exceed chance level in the control setting, in the OOO it achieves a striking 87.67% accuracy, a similar performance as the one reported in Sec. 7.1. On the one hand, this indicates that the model cannot fit the data when the assigned labels are not related to the actual OOO image positions. On the other hand, these results show that the representations learned by LXMERT do encode information regarding the odd-one-out object in the scene.

Taken together, these analyses demonstrate that LXMERT reasons over the multi-image scene to perform the sentence-verification task. In particular, it is able to compute the contextually-defined size of the objects in the scene and perform higher-level reasoning over these representations.

8 Conclusion

We performed an in-depth analysis of the representations learned by the pretrained multimodal transformer LXMERT when performing relational reasoning. We proposed a multimodal reasoning task that requires multi-step relational reasoning and showed that LXMERT can perform the task with high accuracy. Our analysis reveals that the

[14] Implemented in linear support vector machine classification (`LinearSVC`) from the `sklearn`.

majority of the errors arise from target objects with contextually-defined sizes close to the threshold, and that LXMERT solves the task by (i) encoding information regarding the size of objects and by (ii) reasoning over that size. Most of its errors concern *borderline* cases for which the first, image-level reasoning step was shown to be challenging. Overall, our results show that transformer-based architectures pretrained on natural images can generalise to synthetic datasets. We leave to future work an extensive exploration of the extent to which our findings apply to similar tasks and models, for example other vision and langauge transformers (Bugliarello et al., 2021), as well as to natural multimodal data.

Acknowledgments

The authors would like to thank Elia Bruni for providing feedback on a preliminary version of this work and Dieuwke Hupkes for her advice on probing methods. The work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819455).

References

Panos Achlioptas, Judy Fan, Robert Hawkins, Noah Goodman, and Leonidas J Guibas. 2019. Shapeglot: Learning language for shape differentiation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8938–8947.

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *International Conference on Learning Representations (ICLR) – Workshop Track*.

Jacob Andreas and Dan Klein. 2016. Reasoning about pragmatics with neural listeners and speakers. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182.

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Neural module networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 39–48.

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.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Yonatan Belinkov and James Glass. 2019. Analysis methods in neural language processing: A survey. *Transactions of the Association for Computational Linguistics*, 7:49–72.

Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

Emanuele Bugliarello, Ryan Cotterell, Naoaki Okazaki, and Desmond Elliott. 2021. Multimodal pretraining unmasked: A meta-analysis and a unified framework of vision-and-language BERTs. *Transactions of the Association for Computational Linguistics*.

Reuben Cohn-Gordon, Noah Goodman, and Christopher Potts. 2018. Pragmatically informative image captioning with character-level inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 439–443.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Maxwell Forbes, Christine Kaeser-Chen, Piyush Sharma, and Serge Belongie. 2019. Neural naturalist: Generating fine-grained image comparisons. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 708–717.

Noah D Goodman and Michael C Frank. 2016. Pragmatic language interpretation as probabilistic inference. *Trends in cognitive sciences*, 20(11):818–829.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. 2017. Learning to reason: End-to-end module networks for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 804–813.

Dieuwke Hupkes, Sara Veldhoen, and Willem Zuidema. 2018. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926.

Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2018. Learning to describe differences between pairs of similar images. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4024–4034.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Alexander Kuhnle and Ann Copestake. 2017. Shape-World: A new test methodology for multimodal language understanding. *arXiv preprint arXiv:1704.04517*.

Mateusz Malinowski and Mario Fritz. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in neural information processing systems*, pages 1682–1690.

Dong Huk Park, Trevor Darrell, and Anna Rohrbach. 2019. Robust change captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4624–4633.

Sandro Pezzelle and Raquel Fernández. 2019. Is the red square big? MALeViC: Modeling adjectives leveraging visual contexts. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2858–2869.

Yue Qiu, Yutaka Satoh, Ryota Suzuki, Kenji Iwata, and Hirokatsu Kataoka. 2020. 3d-aware scene change captioning from multiview images. *IEEE Robotics and Automation Letters*, 5(3):4743–4750.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Philippe E Ruiz. 2011. Building and solving odd-one-out classification problems: A systematic approach. *Intelligence*, 39(5):342–350.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976.

Lauren A Schmidt, Noah D Goodman, David Barner, and Joshua B Tenenbaum. 2009. How tall is tall? Compositionality, statistics, and gradable adjectives. In *Proceedings of the 31st annual conference of the cognitive science society*, pages 2759–2764. Citeseer.

Jong-Chyi Su, Chenyun Wu, Huaizu Jiang, and Subhransu Maji. 2017. Reasoning about fine-grained attribute phrases using reference games. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 418–427.

Alane Suhr, Mike Lewis, James Yeh, and Yoav Artzi. 2017. A corpus of natural language for visual reasoning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 217–223.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. 2019. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6418–6428.

Hao Tan and Mohit Bansal. 2019. LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5103–5114.

Ramakrishna Vedantam, Samy Bengio, Kevin Murphy, Devi Parikh, and Gal Chechik. 2017. Context-aware captions from context-agnostic supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 251–260.

Gilad Vered, Gal Oren, Yuval Atzmon, and Gal Chechik. 2019. Joint optimization for cooperative image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8898–8907.

A Computing infrastructures

We ran all the experiments with LXMERT using Python 3.7 on a computer with Ubuntu 18.04.5

LTS, single GPU Tesla V100-SXM2, and NVIDIA driver 455.38, CUDA 10.1, and 24GB RAM.

For N2NMN, we used a computer cluster with Debian 10, a single GPU GeForce 1080Ti, 11GB GDDR5X, NVIDIA driver 450.80.02, CUDA 11.0, 260GB RAM, and Python 3.6.

B Hyperparameters and training for N2NMN

We performed a parameter search to determine the best values for training N2NMN[15] on the training split of the POS1 dataset[16] for 3000 iterations of batch size 64 for each combination. We experimented with the following parameters: encoder dropout (0, 0.5, 0.8), decoder dropout (0, 0.5, 0.8), weight decay (5e-5, 5e-4), baseline decay (0.8, 0.99), lambda entropy (0.1, 0.01, 0.001). Their best values (corresponding to the best validation accuracy) are shown in Table 5. We trained the final model using these parameters for 14,000 iterations with batch size 64. The training took approximately 4 hours.

encoder dropout	**decoder dropout**	**weight decay**	**baseline decay**	**lambda entropy**
0.8	0.8	5e-5	0.99	0.01

Table 5: Best parameters for N2NMN model, found with a grid search.

C Hyperparameters and fine-tuning for LXMERT

For the fine-tuning of LXMERT, the pre-trained model with standard hyperparameters was used[17], with only the learning rate changed from 1e-5 to 5e-5, since even with these out-of-the-box parameters, it was able to achieve high performance on the given task. We fine-tuned this model with the POS1 training split using early stopping after 12 epochs, with the parameter number of epochs of BertADAM optimizer set to 150, learning rate 1e-5, and batch size 32 (the only difference in the used hyperparameters during the fine-tuning with 3POS1 was in the batch size 64). We validated the model after each epoch, then the best model was selected, which showed the highest validation accuracy during the 12 epochs, and further evaluated on the test split.

The running time of each fine-tuning epoch for the POS1 dataset was 3 minutes, while each epoch of fine-tuning with 3POS1 took around 6 minutes.

[15] https://github.com/ronghanghu/n2nmn

[16] https://github.com/sandropezzelle/malevic

[17] https://github.com/airsplay/lxmert.git