VQA with No Questions-Answers Training

Ben-Zion Vatashsky  Shimon Ullman
Weizmann Institute of Science, Israel
vatashsky@gmail.com, shimon.ullman@weizmann.ac.il

Abstract

Methods for teaching machines to answer visual questions have made significant progress in recent years, but current methods still lack important human capabilities, including integrating new visual classes and concepts in a modular manner, providing explanations for the answers and handling new domains without explicit examples. We propose a novel method that consists of two main parts: generating a question graph representation, and an answering procedure, guided by the abstract structure of the question graph to invoke an extendable set of visual estimators. Training is performed for the language part and the visual part on their own, but unlike existing schemes, the method does not require any training using images with associated questions and answers. This approach is able to handle novel domains (extended question types and new object classes, properties and relations) as long as corresponding visual estimators are available. In addition, it can provide explanations to its answers and suggest alternatives when questions are not grounded in the image. We demonstrate that this approach achieves both high performance and domain extensibility without any questions-answers training.

1. Introduction

Visual question answering is inspired by the remarkable human ability to answer specific questions on images, which may require analysis of subtle cues, along with the integration of prior knowledge and experience. The learning of new visual classes, properties and relations, can be easily integrated into the question-answering process. Humans can elaborate on the answers they give, explain how they were derived, and why they failed to produce an adequate answer. Current approaches to handle VQA by a machine [67, 62, 55, 77, 32] take a different path, where most answering systems are trained directly to select an answer from common answers of a training set, based on fused image features (mostly using a pre-trained CNN [25]) and question features (mostly using an RNN).

The answering approach we take below is the first, as far as we know, that does not rely on any explicit question-answering training. It uses a process composed according to the question’s structure, and applies a sequence of ‘visual estimators’ for object detection and identifying a set of visual properties and relations. Answering by our ‘Understand, Compose and Respond’ (UnCoRd) approach is divided into two stages (illustrated in Figure 2). First, a graph representation is generated for the question, in terms of classes, properties and relations, supplemented with quantifiers and logical connectives. An answering procedure then follows the question graph, and seeks either a single or multiple assignments of the classes, properties and relations in the graph to the image (Section 3.3). The method is modular, extensible and uses intermediate results to provide elaborated answers, including alternatives to answers not grounded in the image, and notifying about unsupported categories. With an ability to handle extended domains, the UnCoRd approach demonstrates the potential to build a general answering scheme, not coupled to a specific dataset.

Our work includes several novel contributions. First, a method that produces state-of-the-art results on the CLEVR dataset [30] without any questions-answers training. Second, we developed sequence-to-sequence based method, in-
dependent of images, to map questions into their graph representation. Third, we describe a formalism to represent a broad range of possible questions, with an algorithm that finds valid assignments of a question graph in the image, and provides an answer. Fourth, we present a model that can both perform well on CLEVR, as well as generalize to novel domains by just adding visual estimators (for objects, properties and relations) but without QA examples. Some examples are shown in Figure 1 (elaborated later in text).

2. Related Work

Current answering schemes are dominated by end-to-end methods, trained as multi-class classifiers. Many recent works focused on improving the image-question fused features [17, 9, 84, 10, 16], attention mechanisms for selecting important features [80, 54, 47, 4, 12, 59, 50, 27], including self and guided attention [83, 21, 82], applying pre-trained networks [65, 46, 86], and incorporating outputs of other visual tasks [23, 3, 15, 68, 40, 75, 33, 28]. Some provide reasoning using "facts" extraction (e.g. scene type) [73], image caption [42, 1, 41] or by linking visual "facts" with question’s logical form [51, 36]. Other integrated external prior knowledge, by generating a query to a knowledge database [72, 14], fusing it in the representation [78, 58], using a textual image description [39] or by added loss terms [66]. The language prior was addressed as well [22, 20, 13, 76, 57].

Some methods use dynamic networks with architecture affected by the question [18, 56]. The Neural Module Networks (NMN) are dynamically composed out of sub-modules. Originally modules arrangement was based on the dependency parsing of the question [6, 5], while later versions used supervised answering program learning [31, 26, 52, 63], including a probabilistic model [71]. Note that the modules are trained only as components of an answering network for a specific dataset and do no function as independent visual estimators. One method [81] performs full scene analysis in order to carry out the program. This method uses questions-answers training to learn the programs, hence cannot be extended by a simple addition of visual estimators. Moreover, performing full scene analysis (detecting all objects, properties and relations in the scene) may become infeasible for data less restricted than CLEVR (especially for relations). In our method, the answering process is guided by the question and does not perform a full scene analysis. It allows a flexible integration of additional visual capabilities (e.g. novel object classes), providing elaborated answers and proposing alternatives. These capacities are obtained without requiring any QA examples.

Current methods fit models to particular datasets and exploit inherent biases, which can lead to ignoring parts of the question/image, and to failures on novel domains and rephrasing [2, 60]. In contrast to the modular approach we pursue, any adaptation or upgrade requires a full retraining.

3. Method

3.1. Overview

In the formalism we use, a simple question without quantifiers can be transformed to an assertion about the image that may have free variables (e.g., ‘color’ in ‘what is the color of...’). The question is answered by finding an assignment to the image that will make the statement true, and retrieving the free variables. The quantifiers derived from the question require multiple true assignments (such as ‘5’, ‘all’, etc.). The procedure we use seeks the required assignments and
returns the desired answer. The answering process consists of two stages (see Figure 2 for a scheme):

1. **Question mapping into a graph representation** - First, a representation of the question as a directed graph is generated, where nodes represent objects and edges represent relations between objects. Graph components include objects classes, properties and relations. The node representation includes all the object visual requirements needed to answer the question, which is a combination of the following (see examples in the supplement, section 1):
   - **Object class** $c$ (e.g. 'horse').
   - **Object property** $p$ (e.g. 'red').
   - **Queried object property** $f$ (e.g. 'color').
   - **Queried set property** $g$ (e.g. 'number').
   - **Quantifiers** (e.g. 'all', 'two').
   - **Quantity relative to another node** (e.g. same).
   - **Node type**; regular or SuperNode: OR of nodes (with optional additional requirements).

2. **Answering procedure** - In this stage, a recursive procedure finds valid assignments of the graph in the image. The number of required assignments for each node is determined by its quantifiers. The procedure follows the graph, invoking relevant sub-procedures and integrates the information to provide the answer. Importantly, it depends only on the abstract structure of the question graph, where the particular object classes, properties and relations are parameters, used to apply the corresponding visual estimators (e.g. which property to extract). The invoked sub-procedures are selected from a pool of the following basic procedures, which are simple visual procedures used to compose the full answering procedure:
   - Detect object of a certain class $c$.
   - Check the existence of object property $p$.
   - Return an object property of type $f$.
   - Return an object's set property of type $g$.
   - Check the existence of relation $r$ between two objects.

Our construction of a question graph and using its abstract structure to guide the answering procedure leads to our ability to handle novel domains by adding visual estimators but using the same answering procedure. In our method we only train the question-to-graph mappers and the required visual estimators. Unlike QA training, we use independent trainings, which may utilize existing methods and be developed separately. This also simplifies domain extension (e.g. automatic modification is simpler for question-graph examples than for question-image-answer examples).

### 3.2. Question to Graph Mapping

Understanding natural language questions and parsing them to a logical form is a hard problem, still under study [29, 7, 74, 11, 58]. Retrieving question's structure by language parsers was previously performed in visual question answering [6], utilizing the Stanford Parser [34].

We handled the question-to-graph task as a translation problem from natural language questions into a graph representation, training an LSTM based sequence to sequence models [64]. The graph was serialized (using DFS traversal) and represented as a sequence of strings (including special tokens for graph fields), so the model task is to translate the question sequence into the graph sequence (see examples in Section 1 of the supplement). All our models use the architecture of Google’s Neural Machine Translation model [79], and are trained using tensorflow implementation [48]. A simple post-processing fixes invalid graphs. The description below starts with a question-to-graph model trained for CLEVR data, and then elaborates on the generation of extended models, trained for extended scopes of questions.

#### 3.2.1 Question-to-Graph for CLEVR Data

Our basic question-to-graph model is for CLEVR questions and categories (3 objects, 12 properties, 4 property types, 4 relations). The graph annotations are based on the CLEVR answering programs [30], corresponding to the dataset’s questions. The programs can be described as trees, where nodes are functions performing visual evaluations for object classes, properties and relations. These programs can be transferred to our graph representation, providing annotations for our mappers training. Note that concepts may be mapped to their synonyms (e.g. 'ball' to 'sphere').

#### 3.2.2 Extended Question-to-Graph Domain

CLEVR questions are limited, both in the used categories and in question types (e.g. without quantifiers). To handle questions beyond the CLEVR scope, we trained question-to-graph mappers using modified sets of questions (randomization was shown to enable domain extension [69]). There were two types of modifications: increasing the vocabulary of visual elements (object classes, properties and relations) and adding questions of new types. The vocabulary was expanded by replacing CLEVR visual elements with ones from a larger collection. This operation does not add question examples to the set, but uses the existing examples with replaced visual elements. Note that at this stage deals with question mapping and not question answering, the questions, which are generated automatically, do not have to be meaningful (e.g. 'What is the age of the water?') as long as they have a proper mapping, preserving the role of each visual element. To guarantee graph-question correspondence a preprocessing is performed where for each concept, all its synonyms are modified to one form. In addition, for each question all appearances of a particular visual element are replaced with the same term. We used three replacement ‘modes’, each generating a modified dataset by selecting from a corresponding set (real world categories from existing datasets): 1) **Minimal**: Most categories are from COCO [43] and VRD [45] (100 objects, 32 properties, 7 property
types, 82 relations). ii) **Extended**: 'Minimal' + additional categories, sampled from 'VG' (230 objects, 200 properties, 53 property types, 160 relations). iii) **VG**: The categories of the Visual Genome dataset [35] (65,178 objects, 53,498 properties, 53 property types, 47,448 relations, sampled according to prevalence in the dataset). The categories include many inaccuracies, such as mixed categories (e.g. 'fat fluffy clouds') and irrelevant concepts (e.g. objects: 'there are white'), which adds inconsistency to the mapping.

The second type of question modification increased the variability of questions. We created enhanced question sets where additional examples were added to the sets generated by each replacement mode (including 'None'). These examples include questions where 'same <p>' is replaced with 'different <p>' (where <p> is a property), questions with added quantifiers ('all' and numbers) and elemental questions (with and without quantifiers). The elemental questions were defined as existence and count questions for: class, class and property, class and 2 properties, 2 objects and a relation, as well as queries for objects class (in a relation) and property types (including various WH questions).

The words vocabulary we used for training all sets was the same: 56,000 words, composed by the union of the English vocabulary from IWSLT'15 [49] together with all the used object classes, properties and relations. Both the question and the graph representations were based on the same vocabulary, with additional tokens in the graph vocabulary to mark graph nodes and fields (e.g. `<NewNode>`, `<p>`).

Different mappers were trained for all the modified sets above. An example of a graph, mapped using the 'Extended-Enhanced' model, as well as the corresponding original question is given in Figure 3. Note that the modified question, although meaningless, has the same structure as the original question and is mapped to the same graph, except for the replaced visual elements and added quantifiers. This means that the same answering procedure will be carried out, fulfilling our intent to apply the same procedure to similar structured questions.

![Figure 3. Left: A CLEVR question and a corresponding graph. Right: A modified question and a corresponding graph, mapped using Extended-Enhanced model. The accuracy of the modified representation is confirmed, as it matches the original accurate graph (with modified graph concepts).](image)

### 3.3. Answering Procedure

In this stage a recursive procedure seeks valid assignments (see Section 3.1) between the question graph and the image. The question graph, the image and the mask R-CNN [24] produced for the image provide the input to the procedure that recursively processes each node (see Figure 2). For each node, basic procedures (see Section 3.1) are invoked sequentially, according to the node’s requirements and activate visual estimators according to the particular visual elements. The number of required valid assignments is set by the node’s quantifier (a single assignment, a specific number, or all) or by the need of all objects for evaluating the entire object set (e.g. counting, number comparisons). The next processed nodes are according to a DFS traversal. Each basic procedure provides an answer, used to produce the final answer, reporting unsupported categories and providing elaborations, based on intermediate results. For more details and examples see Section 2 of the supplement.

#### 3.3.1 CLEVR Visual Estimators

In order to find a valid assignment of a question graph in the image, and provide the answer, corresponding visual estimators need to be trained. Object locations are not explicitly provided for CLEVR data, however they can be automatically recovered using the provided scene annotations. This process provided approximated contour annotations for CLEVR objects (see Figure 4), which were used for training. Mask R-CNN [24] was used for instance segmentation. For property classifiers, simple CNN models (3 convolutional layers and 3 fully connected layers) were trained to classify color and material; size was estimated according to object’s bottom coordinates and its largest edge. Relations are classified according to objects’ locations.

#### 3.3.2 Real World Visual Estimators

Handling questions in the real-world domain beyond CLEVR objects was performed by utilizing existing visual estimators. For instance segmentation we use a pre-trained mask R-CNN [24] for the 80 classes of COCO dataset [43]. Any other visual estimator may be incorporated to enhance answering capability. In our experiments (Section 4.2.5 and Figure 1) we use color map estimation [70], age and gender classification [37] (utilizing face detection [53]) and depth estimation [44] (utilized for estimating spatial relations).

### 4. Experiments

The experiments tested the abilities of the UnCoRd system, to first, provide accurate results for the CLEVR dataset and second, to handle extended questions and real-world domains. Our analysis included the two answering stages: creating a correct graph representation of the question, and answering the questions. Adam optimizer was used for question-to-graph and visual estimators training with a learning rate of $10^{-4}$ ($10^{-3}$ for the 'Extended-Enhanced' model), selected according to the corresponding validation set results. Each model training was using one NVIDIA
4.1. CLEVR Experiments

We trained a question-to-graph model for CLEVR ("None"-"Basic", as denoted in Section 4.2.1), which generated 100% perfect graphs on its validation set. The visual estimators, described in Section 3.3.1 were also trained and provided the results given in Table 1. CLEVR relations were estimated by simple rules using the objects’ coordinates.

![Instance segmentation example for CLEVR data. Left: GT (approximated from scene data), Right: results.](image)

Table 1. CLEVR estimators results on CLEVR validation set

<table>
<thead>
<tr>
<th>Estimator</th>
<th>P@IoU=0.50</th>
<th>Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ins. seg.</td>
<td>99.9</td>
<td>99.9</td>
</tr>
<tr>
<td>Color</td>
<td>99.9</td>
<td>99.9</td>
</tr>
<tr>
<td>Material</td>
<td>99.9</td>
<td>99.9</td>
</tr>
<tr>
<td>Size</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 2. CLEVR QA accuracy for state-of-the-art methods

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IEP-strong [31]</td>
<td>97.1</td>
<td>92.7</td>
<td>98.7</td>
<td>98.1</td>
<td>98.9</td>
<td>96.9</td>
<td></td>
</tr>
<tr>
<td>FILM [56]</td>
<td>99.3</td>
<td>94.3</td>
<td>93.4</td>
<td>99.3</td>
<td>99.3</td>
<td>97.6</td>
<td></td>
</tr>
<tr>
<td>DDHprog [63]</td>
<td>98.8</td>
<td>96.5</td>
<td>98.4</td>
<td>99.1</td>
<td>99.0</td>
<td>98.3</td>
<td></td>
</tr>
<tr>
<td>MAC [27]</td>
<td>99.5</td>
<td>97.1</td>
<td>99.1</td>
<td>99.5</td>
<td>99.5</td>
<td>98.9</td>
<td></td>
</tr>
<tr>
<td>TDo [52]</td>
<td>99.2</td>
<td>97.6</td>
<td>99.4</td>
<td>99.5</td>
<td>99.6</td>
<td>99.1</td>
<td></td>
</tr>
<tr>
<td>HAN [50]</td>
<td>99.6</td>
<td>97.2</td>
<td>96.9</td>
<td>99.6</td>
<td>98.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NS-VQA [81]</td>
<td>99.9</td>
<td>99.7</td>
<td>99.9</td>
<td>99.8</td>
<td>99.8</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

UnCoRdNone-B 98.99 99.54 99.91 99.74 99.80 99.74 99.8

Table 3. Accuracy of question-to-graph mapping for all data types

As can be seen, our model achieves state-of-the-art results without training for the visual question answering task and not using any answers GT, as other methods. In addition, UnCoRd can elaborate and explain answers and failures using intermediate results, and extend the handled domain with no need of images and related QA examples, as demonstrated in Section 4.2 and Figure 6. On a sample of 10,000 validation set examples, all mistakes were due to wrong visual estimators’ predictions, mainly miss-detection of a highly occluded object. Hence, accurate annotation of object coordinates (as performed in NS-VQA [81]) may even further reduce the small number of errors. Note that NS-VQA requires full scene analysis, which is not scalable for domain extension with a large number of objects and relations. It also uses images with question-answer pairs to train the programs, coupling the method to the specific

trained question answering domain.

4.2. Out of Domain Experiments

Next, we test UnCoRd beyond the scope of CLEVR data. We trained question-to-graph models on the modified and enhanced CLEVR data and used corresponding visual estimators. We examined whether domain extension is possible while maintaining a good performance on the original data.

4.2.1 Question to Graph

For evaluating question representation, we trained and tested (see Section 3.2.2) 8 question-to-graph models that include all replacement modes (None, Minimal, Extended, VG), each trained in two forms: Basic (B), i.e. no added question examples (∼700K examples) and Enhanced (E), i.e. with additional examples (∼1.4M examples).

In Table 3, we report the results of each trained model on the validation sets of all 8 models, which provides information on generalization across the different sets. Note that as the "None" extension, unlike the data of other models, includes mapping from concepts to their synonyms (see Section 3.2.2), prediction for "None" data by the "Minimal", "Extended" and "VG" models include a preprocessing stage transforming each concept synonyms to a single form.

Results demonstrate that models perform well on data with lower variability than their training data. The high performance of the 'Extended' models on their corresponding data illustrates that substantial extensions are possible in question-to-graph mapping without requiring any new training images. VG models’ lower accuracy is expected due to the unsuitable elements in its data (see Section 3.2.2). Additional tests are required to check possible advantages of VG models for different domains. We report such a test next.

4.2.2 VQA Representation

In this experiment, representation capabilities are tested for a different dataset. Since normally, annotations corresponding to our graph representation are not provided, we sampled 100 questions of the VQA [8] validation set and manually examined the results for the eight question-to-graph models (see Section 4.2.1).

The results in Table 4 express the large gaps in the abilities of models to represent new domains. Models trained
specifically on CLEVR do not generalize at all to the untrained domain. As the models are trained on more diverse data, results improve substantially, peaking clearly for VG-Enhanced model by a large margin from other models. This is also evident in the example given in Figure 5 where adequacy of the graph increases in a similar manner. This result is interesting as using this model provides high accuracy for CLEVR as well (see Table 5). The fact that substantial performance gain is achieved for a data domain that was not used in training (the VQA dataset domain), while preserving good results on the original data (CLEVR), demonstrates the potential of the approach to provide a general answering system for visual questions. Further investigation is required for means to enrich question description examples and produce further significant improvements.

![Figure 5. Generated graphs for a free form question](image)

Table 4. Accuracy of graph representation for VQA [8] sample, given for the different UnCoRd mappers. As expected, training on more diverse data allows better generalization across domains.

<table>
<thead>
<tr>
<th>Mapper</th>
<th>Exist</th>
<th>Count</th>
<th>Comp. Num.</th>
<th>Query Att.</th>
<th>Comp. Att.</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>E</td>
<td>B</td>
<td>E</td>
<td>B</td>
<td>E</td>
<td>B</td>
</tr>
<tr>
<td>None</td>
<td>0</td>
<td>12</td>
<td>22</td>
<td>22</td>
<td>34</td>
<td>50</td>
</tr>
<tr>
<td>Minimal</td>
<td>1</td>
<td>12</td>
<td>22</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extended</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Figure 6. Examples for answering different question types on CLEVR images](image)

4.2.3 Maintaining Performance on CLEVR Questions

We evaluated the performance change for the CLEVR test set, as the training data variability of the question-to-graph models increases. The results are given in Table 5.

<table>
<thead>
<tr>
<th>Mapper</th>
<th>Exist</th>
<th>Count</th>
<th>Comp. Num.</th>
<th>Query Att.</th>
<th>Comp. Att.</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>B</td>
<td>E</td>
<td>99.89</td>
<td>99.91</td>
<td>99.91</td>
<td>99.91</td>
</tr>
<tr>
<td>E</td>
<td>99.94</td>
<td>99.80</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimal</td>
<td>B</td>
<td>E</td>
<td>99.89</td>
<td>99.91</td>
<td>99.91</td>
<td>99.91</td>
</tr>
<tr>
<td>E</td>
<td>99.94</td>
<td>99.80</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extended</td>
<td>B</td>
<td>E</td>
<td>99.82</td>
<td>99.91</td>
<td>99.91</td>
<td>99.91</td>
</tr>
<tr>
<td>E</td>
<td>99.94</td>
<td>99.80</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VG</td>
<td>B</td>
<td>E</td>
<td>98.03</td>
<td>97.97</td>
<td>97.62</td>
<td>97.22</td>
</tr>
<tr>
<td>E</td>
<td>99.70</td>
<td>99.54</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Figure 7. Examples for CLEVR-Humans questions](image)

4.2.4 CLEVR Humans

An example of using the CLEVR images with different questions is the CLEVR-Humans [31] (7145 questions in test set), where people were asked to provide challenging questions for CLEVR images. The questions vary in phrasing and in the required prior knowledge.

<table>
<thead>
<tr>
<th>Method</th>
<th>No FT</th>
<th>FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>IEP-18k</td>
<td>54.0</td>
<td>66.6</td>
</tr>
<tr>
<td>FILM</td>
<td>56.6</td>
<td>75.9</td>
</tr>
<tr>
<td>MAC</td>
<td>57.4</td>
<td>81.5</td>
</tr>
<tr>
<td>NS-VQA</td>
<td>-</td>
<td>67.0</td>
</tr>
</tbody>
</table>

Table 5. Accuracy of CLEVR dataset question answering by UnCoRd using the different question-to-graph mappers.

It is evident that even models that were trained on a much larger vocabulary and question types than the original CLEVR data still perform well, mostly with only minor accuracy reduction. This demonstrates that with more variable training we can handle more complex questions, while maintaining good results on the simpler domains. Examples on CLEVR images for both CLEVR questions and others are shown in Figure 6 (using ‘None-Enhanced’ mapper).
Results, given in Table 6, demonstrate that for models without finetuning, our 'None-Enhanced' model provides state-of-the-art results (without any answer examples). For all models, questions with phrasing not included in training are prone to errors, including 'ballucinations' of concepts. Note that CLEVR-Humans answers are the same answers as in CLEVR (by instructions to workers), hence models biased towards CLEVR (the "None" models) have a better success chances. Models with a rich vocabulary may capture the question graph more accurately, but that may include concepts with no corresponding visual estimators, resulting with answers such as: "Unknown relation 'in between'". Adding such visual estimators will improve performance. Since accuracy calculation does not reward for such limitation indications, just "guessing" the answer would increase the computed accuracy, especially as success chances rise with a simple question categorization (e.g. 50% for yes/no and size questions). However, indicating limitations gives a better sense of the system's level of understanding the question, and can lead to corrective actions. Such answers can be promoted in QA systems, by reducing "score" for wrong answers, or giving partial scores to answers identifying a missing component.

Examples of CLEVR-Humans questions are given in Figure 7. It is evident that the more general model (VG-Enhanced) can perform on out of scope questions (top) and report limitations (bottom).

4.2.5 Extensibility to Real-World Images

The UnCoRd system can be naturally extended to novel domains by a simple plug-in of visual estimators. This is illustrated in Figure 1 for using new properties/relations and for an entirely different domain of real-world images. An experiment that adds questions with a novel property is presented in Section 3 of the supplement. We next describe an experiment for real-world images, where we use real world visual estimators (see Section 3.3.2) and our most general trained mapper (VG-Enhanced). We compare our model to Pythia [85], which has top performance on the VQA v2 dataset [19]. The experiment includes two parts:

1. 'Non VQA_v2' questions: 100 questions outside Pythia’s training domain (VQA v2), with unambiguous answers, on 50 COCO images (two similar questions per image with different answers). We freely generated questions to include one or more of the following categories:
   - A combination of properties and relations requirements linked by logical connectives ('and', 'or').
   - Property comparison (e.g. 'same color').
   - Quantifiers (e.g. 'all', 'five').
   - Quantity comparison (e.g. 'fewer', 'more').
   - A chain of over two objects connected by relations.

2. 'VQA_v2' questions: 100 questions sampled from VQA v2 dataset [19] with terms that have visual estimators in UnCoRd and unambiguous answers (annotated by us).

In addition to the estimators mentioned in Section 3.3.2, ConceptNet [61] is used by UnCoRd to query for optional classes when superordinate groups are used (e.g. 'animals'). More details are in Section 4 of the supplement.

The non VQA_v2 results, given in Table 7, demonstrate the substantial advantage of UnCoRd for these types of questions. All UnCoRd’s failures are due to wrong results of the invoked visual estimators. Note the substantial performance difference in Pythia between yes/no and WH questions, unlike the moderate difference in UnCoRd. We found that Pythia recognizes the yes/no group (i.e. answers 'yes'/'no'), but its accuracy (56%) is close to chance level (50%). Examples of successful UnCoRd answers to the non VQA_v2 questions are provided in Figure 8, while failure examples, including failure sources, are shown in Figure 9. Pythia’s answers are given as well.

<table>
<thead>
<tr>
<th>Method</th>
<th>Yes/No</th>
<th>WH</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pythia [85]</td>
<td>56.0</td>
<td>14.0</td>
<td>35.0</td>
</tr>
<tr>
<td>UnCoRd-VG-E</td>
<td>88.0</td>
<td>64.0</td>
<td>76.0</td>
</tr>
</tbody>
</table>

Table 7. Answering accuracy for 100 questions outside the VQA v2 domain (including quantifiers, comparisons, multiple relation chains and multiple relations and properties) on COCO images.

Figure 8. Examples of UnCoRd successes in answering questions outside the VQA v2 domain on COCO images.

Results for the 100 VQA_v2 questions are given in Table 8. As can be seen, UnCoRd’s results are better by a large margin, compared to Pythia [85] end-to-end model, even
though questions were sampled from VQA v2, a dataset used for Pythia’s training. As in the previous part, all UnCoRd’s failures are only due to wrong results of the invoked visual estimators. Examples of UnCoRd’s answers for the VQA v2 questions are given in Figure 10, including the corresponding answers of Pythia.

<table>
<thead>
<tr>
<th>Method</th>
<th>Yes/No</th>
<th>WH</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pythia [85]</td>
<td>90.0</td>
<td>68.3</td>
<td>77.0</td>
</tr>
<tr>
<td>UnCoRd-VG-E</td>
<td>97.5</td>
<td>88.3</td>
<td>92.0</td>
</tr>
</tbody>
</table>

Table 8. Answering accuracy for 100 questions sampled from VQA v2 dataset (on terms with visual estimators in UnCoRd).

The above experiments on real-world images show that when corresponding visual estimators are available, our method performs better than a leading end-to-end model, both for questions outside the training domain of the end-to-end model (where the advantage is substantial) and for questions from this domain. This is achieved without any question answering training.

5. Conclusions and Future Directions

We proposed a novel approach to answer visual questions by combining a language step, which maps the question into a graph representation, with a novel algorithm that maps the question graph into an answering procedure. Because the algorithm uses the abstract structure of this graph, it allows a transfer to entirely different domains. Training is performed for the language step to learn the graph representation, and for the visual step to train visual estimators. However, unlike existing schemes, our method does not use images and associated question-answer pairs for training. Our approach allows handling novel domains provided that corresponding visual estimators are available. The combination of the question graph and answering procedure also gives the method some capacity to explain its answers and suggest alternatives when question is not grounded in the image. Based on this approach, our answering system achieves near perfect results on a challenging dataset, without using any question-answer examples. We have demonstrated that question representation and answering capabilities can be extended outside the scope of the data used in training, preserving good results for the original domain.

Substantial work is required to obtain a system that will be able to perform well on entirely general images and questions. The main immediate bottleneck is obtaining question-to-graph mapping with general representation capabilities for a broad range of questions. Question graph representation may also be enhanced to support questions with more complex logic, as well as extending the scope of the supported visual categories (e.g., global scene types). Any general VQA requires vast estimation capabilities, as any visual category can be queried. In UnCoRd they are modularly incremented and automatically integrated with existing questions. Additional basic areas that current schemes, including ours, have only begun to address, are the use of external, non-visual knowledge in the answering process, and the composition of detailed, informative answers, integrating the language and visual aspects of VQA.

Acknowledgements: This work was supported by EU Horizon 2020 Framework 785907 and ISF grant 320/16.
References


