Supplementary Note: Visual Question Answering based on Formal Logic

I. Target Sentence to Rule Conversion

The target sentence defined in section 2.2 is constructed such that the break in sentence (BIS) character, "\" , signifies the change in the rule. For example, in the sentence

$$\text{attribute(X, shape, cube), attribute(X, color, green)} \\ \backslash\text{attribute(Y, color, blue), attribute(Y, material, metal)}$$

While converting the above sentence into a rule, the predicates leading upto the BIS character form the first rule. The same process is repeated on the remaining sentence until the end of the sentence is reached. Once the individual rules are separated and each rule is assigned a head predicate $\mathsf{r}_i(\cdot)$ where i is the rule number and $i = 1, \ldots, p-1$ and p is the total number of rules. The last rule is called the target rule as we obtain the required answer from it. The argument of the head predicate is taken as the first argument of the last predicate in the body of the rule. In the above example, the first rule would be written as,

$$\mathsf{r}_1(\mathsf{X}) \leftarrow \text{attribute(X, shape, cube), attribute(X, color, green).}$$

The count operation is encoded using the character C_i, where $i \in \mathbb{N}$. Moreover, it is always sandwiched between two BIS characters in the target sentence. During the conversion to rules, the count character is replaced by the count predicate with the previous rule head as its first argument. Consider the following sentence,

$$\text{attribute(X, shape. cube), attribute(X, color, green), relation(Y, X, left)}\backslash\ \mathsf{C}_1 \backslash, \ldots$$

In this case the converted rules would be,

$$\begin{aligned} \mathsf{r}_1(\mathsf{Y}) \leftarrow & \text{attribute(X, shape. cube),} \\ & \text{attribute(X, color, green), relation(Y, X, left).} \\ \mathsf{r}_2(\mathsf{C}) \leftarrow & \text{count}(\mathsf{r}_1(\mathsf{Y}), \text{C}). \\ & \vdots \end{aligned}$$

Finally, disjunction operation is encoded using the character ";". When ";" is encountered while parsing the sentence, the rule heads of the previous two rules are connected using a disjunction operation "$\vee$". To illustrate this, consider the sentence below

$$\text{attribute(W, shape. cube)}\backslash\text{attribute(X, color, green), relation(Y, X, left)}\backslash;\text{ , attribute(Z, color, red).}$$

This would translate to,

$$\begin{aligned} \mathsf{r}_1(\mathsf{W}) & \leftarrow \text{attribute(W, shape, cube).} \\ \mathsf{r}_2(\mathsf{Y}) & \leftarrow \text{attribute(X, color, green), relation(Y, X, left).} \\ \mathsf{r}_3(\mathsf{Z}) & \leftarrow \big(\mathsf{r}_1(\mathsf{Z}) \vee \mathsf{r}_1(\mathsf{Z})\big), \text{attribute(Z, color, red).} \end{aligned}$$

II. Inconsistent Functional Form in GQA dataset

Our approach relies on having a functional form accompanying the question to generate the target rule for that question. Functional form decomposes each question into a set of operations which when performed on the image results in the desired answer to the question. Each operation in the functional form has 3 parts, namely (i) operation name, (ii) arguments and (iii) dependencies. Consider the question "Is the grass green and tall?" taken from the GQA dataset. It's functional form is given by

```
select: grass (4569011)->verify color:
green [0] ->verify height: tall  [0]->
and:  [1, 2]
```

Here `->` indicates the change in the operation. Table I shows the individual components of the operations in the functional form for the question mentioned above. These operations are then converted to predicates listed in table 1 in the main paper. For additional information on functional form refer to [1]. For operations that depend on two objects, the argument and the dependency of the operation determines the ordering of the arguments in the corresponding predicates. For example, consider the operation `relate: chair, left [1]`, the instruction here is to find the chair to the left of the object that was the output of operation 1. This would be translated to relation(Y, X, left), where Y refers to chair and X refers to object 1.

For some questions in the GQA dataset, we noticed that sometimes there were inconsistencies with the arguments for the `relate` operation when present in the functional form. This lead to the target sentence generated using those functional forms to be incorrect. To illustrate this, consider the question taken from the GQA dataset, "Do you see any bookcase to the left of the napkin the cat is to the right of?". Here, we have three objects, namely, a cat, a napkin and a bookcase. Furthermore, we know that the cat is to the right of the napkin. In shorthand we can denote this

TABLE I
FUNCTIONAL FORM FOR THE QUESTION "IS THE GRASS GREEN AND TALL?"

S.No	Name	Arg	Dep
0	`select`	`grass`	-
1	`verify color`	`green`	[0]
2	`verify height`	`tall`	[0]
3	`and`	-	[1, 2]

physical relation as napkin←cat and we want to know if bookcase→napkin←cat is true. The functional form(taken from the GQA dataset) for this question is given by,

```
select: cat (1298333)->relate: napkin,
to the right of,o (1298346) [0]->relate:
bookcase,to the left of,s (1298370) [1]
->exist: ? [2]
```

Sticking to the convention we defined earlier, if we parse the functional form, we can see that the first `relate` operation is looking for napkins to the right of the cat and the second `relate` operation checks for bookcase to the of napkin. This would result in the wrong answer as the first `relate` operation is checking for the wrong relation.

Say we change our convention and consider `relate: chair, left [1]`, then we try to find the object 1 to the left of the chair. Now, when we parse the above functional form we can see that the first `relate` operation is looking for the right relation (cat to the right of napkin), but the second `relate` operation checks for napkin to the left of the bookcase, which is again incorrect. This implies that no interpretation would yield in the right answer.

Therefore, while generating the training data for the transformer network we only restrict ourselves to questions having consistent functional forms.

References

[1] D. A. HUDSON AND C. D. MANNING, *Gqa: A new dataset for real-world visual reasoning and compositional question answering*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 6700–6709.