Strong Supervision for Visual Questioning Answering on GQA

Ammar Husain
mrahusain@gmail.com

Abstract

Vision-and-language reasoning requires an understanding of visual concepts, language semantics, and, most importantly, the alignment and relationships between these two modalities. Reasoning involves a multi-stage process that humans intuitively carry out. A similar sequence of deduction steps have been introduced in the GQA dataset through Functional Programs that are paired with natural language questions that a visual question-answering system is tasked to answer. We propose several methodologies of integrating these Functional Programs to finetuning a visio-linguistic Transformer based encoder, LXMERT. While the experiments in this paper were carried out with LXMERT, these methodologies of applying Strong Supervision to task finetuning can be used on other Transformer based architectures as well. We provide experimental results on several visual question-answering accuracy metrics and add justifications for why these methodologies do not necessarily boost overall model performance.

1 Introduction

Visual Question Answering tasks, where a system has to answer free form questions in natural language about presented natural images, embodies some of the hardest challenges in artificial intelligence today. The model must not only assimilate knowledge from one stream (visual) and translate it into another (e.g Image Captioning), it must also draw compositional inferences. The task demands a rich set of abilities as varied as object recognition, commonsense understanding and relation extraction, spanning both the visual and linguistic domains.

There has been substantial past works in separately developing backbone models with better representations for the single modalities of vision (Szegedy et al., 2014; Simonyan and Zisserman, 2015) and of language (Devlin et al., 2019; Peters et al., 2018). Leveraging these separate language and vision models pretrained for other large-scale tasks the dominant strategy was to then learn grounding as part of task training. This often resulted in myopic groundings that generalize poorly when paired visiolinguistic data and is limited or biased. To fully exploit recent successes in self-supervised learning that have captured rich semantic and structural information from large, unlabelled data sources research shifted toward cross modality pretraining (Qi et al., 2020; Chen et al., 2020; Tan and Bansal, 2019; Lu et al., 2019). These models develop a common visual grounding that can learn these connections and leverage them on a wide array of vision-and-language tasks by pretraining them to perform so-called 'proxy' tasks. These proxy tasks leverage structure within the data to generate supervised tasks automatically (e.g. colorizing images or reconstructing masked words in text (Devlin et al., 2019)).

1.1 Task

Following the large-scale pre-training, these visiolinguistically grounded models are finetuned on specific tasks usually by building classifiers on the output vectors of the last hidden state or tokens. An especially challenging task, as mentioned before, is visual question answering. There have been several datasets tackling this problem (Agrawal et al., 2016; Goyal et al., 2017; Johnson et al., 2016), with the state of the art in terms of difficulty being GQA (Hudson and Manning, 2019). The questions in GQA involve hierarchical and compositional visual reasoning on a large set of natural images, thereby making it difficult for the model to simply memorize. Each image in the dataset is annotated with a dense scene graph representing the objects, attributes and relations it contains. Each question is associated with structured representations in the

form of functional programs that specify their contents and semantics, and are visually grounded in the image scene graphs.

Several cross modally (vision + language) pretrained models (Chen et al., 2020; Tan and Bansal, 2019; Lu et al., 2019) have demonstrated strong performance on GQA as compared to the CNN+LSTM baselines. However none of these experiments exploit the strong supervision that the GQA dataset provides in terms of the "query-language like" functional programs associated with natural language questions [see Section 2]. These functional programs provide unambiguous reasoning steps that a model/human could incorporate in order to deduce the answer.

The core hypothesis of this paper is that it would help boost model performance on answering unseen visual questions, if the model was trained jointly, in addition to the visual image, with the natural language question as well as its corresponding query-like functional program.

2 Data

This paper uses GQA: A New Dataset for Real-World Visual Reasoning and Compositional Question Answering (Hudson and Manning, 2019) (`visualreasoning.net`), which is a publicly available dataset built for hierarchical and compositional visual reasoning on natural images. The GQA dataset provides 113K images and 22M questions of assorted types and varying degrees of compositionality. 1 provides an example of the images and questions encountered in this dataset.

Figure 1: Example of GQA image-based questions

A1. Is the **tray** on top of the **table** black or light brown? light brown
A2. Are the **napkin** and the **cup** the same color? yes
A3. Is the small **table** both oval and wooden? yes
A4. Is there any **fruit** to the left of the **tray** the **cup** is on top of? yes
A5. Are there any **cups** to the left of the **tray** on top of the **table**? no
B1. What is the brown **animal** sitting inside of? **box**
B2. What is the large **container** made of? cardboard
B3. What **animal** is in the **box**? **bear**
B4. Is there a **bag** to the right of the green **door**? no
B5. Is there a **box** inside the plastic **bag**? no

These images are all derived from the Visual Genome (Krishna et al., 2016) dataset and all images, questions and corresponding answers are accompanied by matching semantic representations. While not specified in the paper, further online research indicates that the hold out test set images for this dataset come from the MS-COCO dataset (Lin et al., 2015).

As mentioned in 1.1, each natural language question is annotated with a query-language like functional program. For example, an image such as 2, would have a natural language question:

> What color is the food on the red object left of the small girl that is holding a hamburger, yellow or brown?

The associated functional program annotation for this question string that deduces the final answer would be:

```
Select: hamburger;
Relate: (girl, holding);
Filter: (size, small);
Relate: (object, left);
Filter: (color, red);
Relate: (food, on);
Choose: (color: yellow | brown)}
```

Figure 2: GQA example image

The authors annotate each question either as a structural or semantic one. Structural questions can be (i) verify for yes/no questions, (ii) query for all open questions, (iii) choose for questions that present two alternatives, (iv) questions involving logical inference, (v) comparison questions with more than two alternatives. The semantic type refers to the main subject of the question: (i) object, (ii) attribute, (iii) category, (iv) relation or (v)

global about overall properties of the scene such as weather or place.

3 Pre-trained Model

To pursue the experiments for this project on joint natural language question + functional program finetuning, we chose the "LXMERT: Learning Cross-Modality En-coder Representations from Transformers" (pronounced: 'leksmert') (Tan and Bansal, 2019) model architecture. LXMERT builds a pre-trained vision-and-language cross-modality framework and shows its strong performance on several datasets. Their framework is modeled after recent BERT-style innovations while further adapted to useful cross-modality scenarios. It consists of three Transformer (Vaswani et al., 2017) encoders: an object relationship encoder, a language encoder, and a cross-modality encoder. In order to better learn the cross-modal alignments between vision and language, LXMERT is pre-trained with five diverse representative tasks: (1) masked cross-modality language modeling, (2) masked object prediction via RoI-feature regression, (3) masked object prediction via detected-label classification, (4) cross-modality matching, and (5) image question answering.

While there were several other pre-trained Transformer based visiolinguistic models (Qi et al., 2020; Chen et al., 2020; Lu et al., 2019) that could potentially be used as well, we chose LXMERT for the following reasons:

- Model was already evaluated on the GQA dataset and was one of the top-5 performers in the 2019 challenge.
- Model and pre-trained weights have been open sourced by the authors with detailed instructions on how to fine tune the base model on GQA. Additionally it has also been integrated into HuggingFace.
- LXMERT architecture is a two stage Transformer with Language Encoder (as compared with UNITER (Chen et al., 2020)). Since the hypothesis only focuses on joint training of the language tokens, fine tuning will not affect the weights of the Object-Relation Encoder.
- The model pre-training for LXMERT used only the Masked Language Modeling (MLM) task objective as introduced in BERT (Devlin et al., 2019) for encoding image captions. It however did not incorporate the Next Segment Prediction (NSP) objective also used in BERT. This rationale will be further elaborated in Section 4.2.1.

4 GQA Finetuning Approach

The key contributions of this project are in the approach to finetuning on the visual question answering task for GQA, through strong supervision. Strong supervision is the strategy to exploit textual functional programs along with natural language questions during model training. The vanilla LXMERT model was trained with just the natural language question as tokenized language input and a multi-class classifier on the hidden state of the cross-modal [CLS] token, as shown in 4 For the experiments in this project we pursue combinations of the following two approaches to inject strong supervision:

4.1 Append Functional Program to Question

This approach (as shows in 5 simply appends the Question and Functional Program together as the language input while fine tuning the model for visual question answering on the GQA dataset. The Language Encoder would then consume:

```
[CLS] Question_tokens
Functional_Program_tokens [SEP]
```

As compared with the baseline LXMERT model fine-tuning that input:

```
[CLS] Question_tokens [SEP]
```

4.2 BERT-like language tasks

Figure 6 illustrate the two 'proxy' task heads described in this subsection.

4.2.1 Q-FPM Objective

This approach attempts to learn the joint probability that a (Question, Functional Program) tuple is valid or not. This is analogous to the Next Segment Prediction (NSP) task from BERT (Devlin et al., 2019). It is also loosely analogous to the Cross-Modality Matching task head for LXMERT pre-training where the model learns to predict whether the image matches the corresponding textual input.

For this experiment, a new task objective, we call Q-FPM was created using a Binary Cross Entropy loss. The objective learned to predict whether the provided Functional Program (FP) matches the provided natural language Question (Q). The task

Figure 3: LXMERT: Pre-training tasks

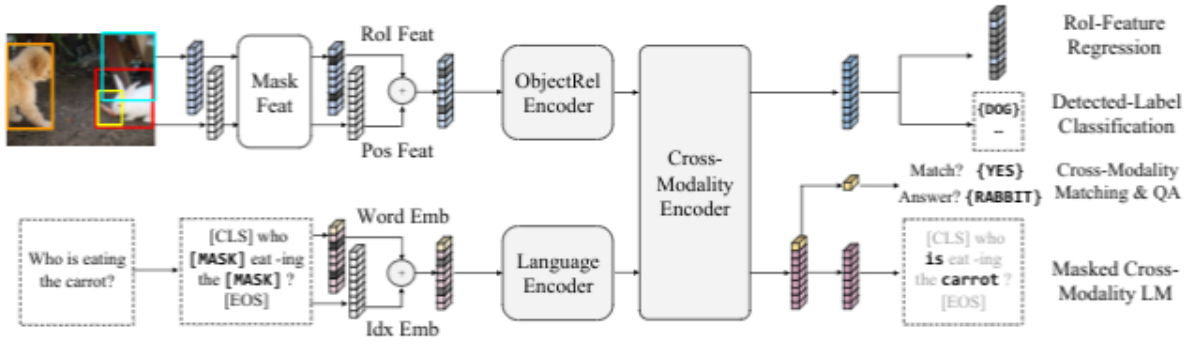
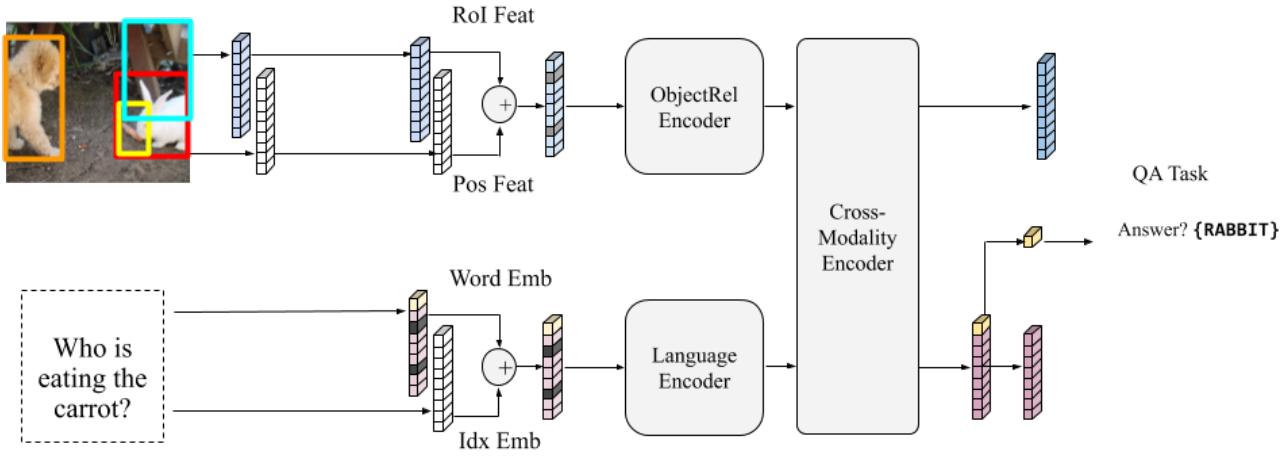


Figure 4: LXMERT: GQA Finetuning



was setup similar to the NSP task in BERT [Devlin et al] with the language input being:

```
[CLS] Question_tokens [SEP]
Functional_Program_tokens [SEP]
```

and the corresponding segment ids were:

```
[BOS] SEGMENT_0 [SEP]
SEGMENT_1 [EOS]
```

The task objective used the final hidden representation of the [CLS] token, passed it through a simple feed-froward layer to get a binary prediction on whether the second segment corresponds to the first segment. The task training sampled 50% negative examples by randomly picking an FP for a different natural language question (Q). Given the diversity of the questions in the dataset the probability of sampling a negative example that could actually be positive is negligible.

4.2.2 Masked Lanuage Model (MLM) Objective

This training objective was exactly the same as introduced in BERT (Devlin et al., 2019), where certain tokens were masked out and the model attempted to predict them from the hidden representation of the final language model output.

5 Metrics

In addition to the conventional metrics measuring overall question answering **Accuracy** on the test dataset, we will use the following additional accuracy metrics to analyze model performance.

- **Binary**: Measures accuracy over answers that involve a binary prediction such as: Yes, No.
- **Open**: Measures accuracy over questions that are open ended, aka multi-class classification problems.
- **Validity**: Checks whether a given answer is

Figure 5: Appending Functional Program to Natural Language Question in the language tokens

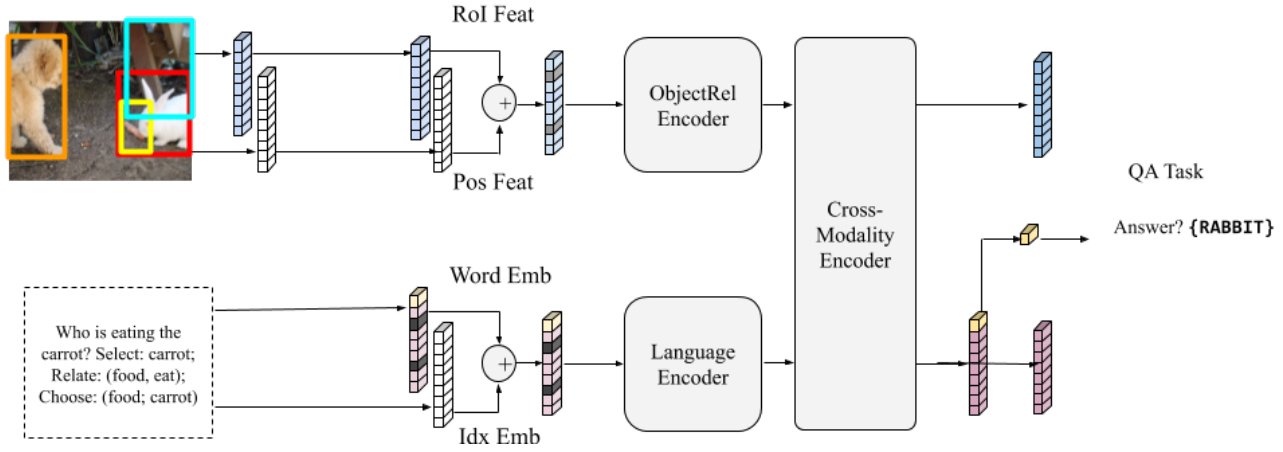
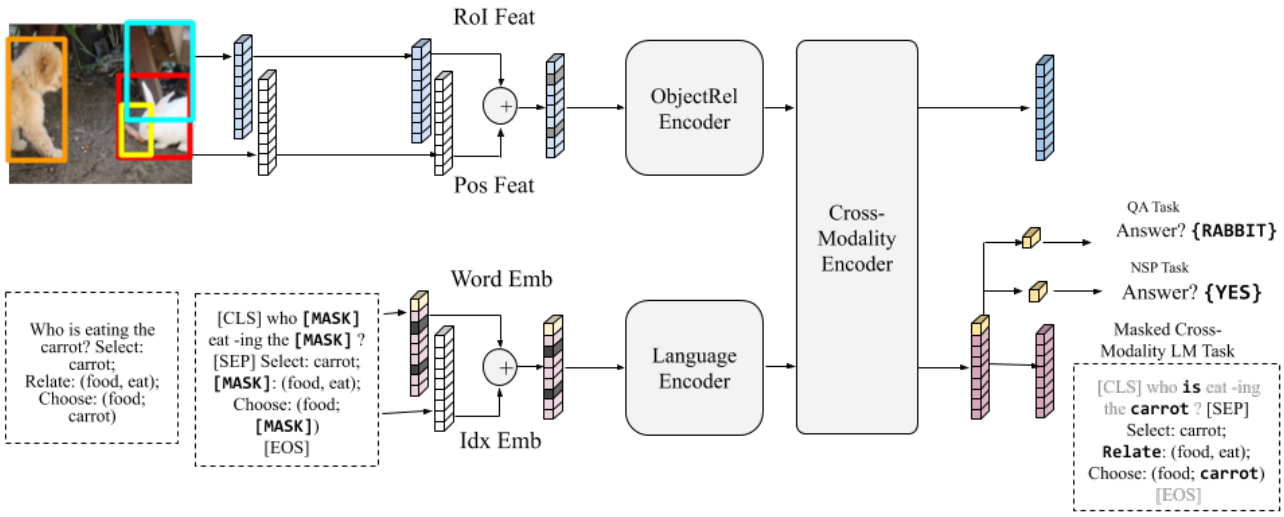


Figure 6: Proxy task heads for GQA finetuning



in the question scope, ex: responding with a color to a question pertaining to color.

- **Plausibility**: Checks how plausible the model provided answer is. For ex: model should answer purple when asked for the color of an apple.

- **Consistency**: Measures that responses are consistent across different questions. For ex: a model should not answer green to a new question about an apple it has just identified as red.

- **Distribution**: Measures the overall match between the true answer distribution and the model predicted distribution. This allows us to see if the model predicts not only the most common answers but also the less frequent ones.

6 Results and Analysis

The following experiments were run to finetune a pretrained LXMERT model on the GQA dataset for 4 epochs with a `MAX_TOKEN_LENGTH` of 200.

- **LXMERT Baseline** Finetuned the baseline model on the GQA dataset without the functional programs.

- **Append FP** Finetuned the baseline LXMERT model on the GQA dataset with the functional programs appended after the natural language question.

- **(Q-FPM, MLM) $\rightarrow$ FT** Ran 4 epochs of pretraining using the proxy Q-FPM and MLM

Output	LXMERT	Append FP	(Q-FPM, MLM) → FT	Joint: (Q-FPM, MLM, FT)	Joint: (Q-FPM, FT)
Accuracy	60.2565	58.9703	57.8112	59.9435	60.2680
Binary	77.2411	74.6192	72.0947	76.1805	76.9872
Open	45.2659	45.1585	45.2045	45.6127	45.5114
Validity	96.3026	96.4314	96.0744	96.2211	96.3058
Plausibility	84.3856	84.5812	84.2209	84.3400	84.4932
Consistency	89.4882	89.2677	88.5118	89.9090	90.1639
Distribution	5.3561	5.5604	5.2998	5.7046	5.3915

Table 1

task heads as described in Sections 4.2.1, 4.2.2 respectively. This was followed by 4 epochs of finetuning with the functional programs appended to the natural language question.

- **Joint: (Q-FPM, MLM, FT)** Ran 4 epochs of finetuning with the functional program on the GQA dataset, jointly with the Q-FPM and MLM task heads. There were 3 loss values that were added together for each optimization step.

- **Joint: (Q-FPM, FT)** Ran 4 epochs of finetuning with the functional program on the GQA dataset, jointly with the Q-FPM task head. There were 2 loss values that were added together for each optimization step.

Table 1 shows the results for the various experiments described above evaluated on the metrics described in Section 5. As is evident from the results adding strong supervision through functional programs with the methods described in Section 4 does not improve performance of the baseline LXMERT model. For the first three approaches it actually hinders the model's ability to perform well on GQA. This is most likely because of the following reasons:

1. The pretraining data involved a huge corpus of image-text pairs, where the text mostly consisted of natural language. Therefore the syntactic and semantic form of the functional programs within GQA was a miniscule signal compared to its prior knowledge.

2. With the approaches in these experiments, the functional programs were not grounded in the images themselves. They were merely attached on to the language encoder and didnt draw connections back into the image regions.

3. The additional task heads served more as a distraction to the core optimization process than a boost to the model's reasoning abilities.

4. Functional programs were not used during test inference and given the lack of grounding of the model, this led to training data bias rather than enhanced reasoning.

7 Conclusion

We presented approaches for introducing strong supervision during finetuning for visual question-answering on the GQA dataset. This paper introduces several approaches for injecting Functional Programs into the Language Encoder for Cross-Modality based Transformer architectures. Additionally, we provide experimental results for these approaches by finetuning the LXMERT pre-trained model and evaluating on several accuracy metrics for visual question answering. Finally we present a list of drawbacks rationale on why the approaches described in this paper do not provide the intended and statistically significant performance boost.

References

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. 2016. Vqa: Visual question answering.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering.

Drew A. Hudson and Christopher D. Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2016. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft coco: Common objects in context.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. 2020. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. Going deeper with convolutions.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers.