Answer-Me: Multi-Task Open-Vocabulary Learning for Visual Question-Answering

AJ Piergiovanni, Wei Li, Weicheng Kuo, Mohammad Saffar, Fred Bertsch and Anelia Angelova
Google Research

Abstract

We present Answer-Me, a task-aware multi-task framework which unifies multiple question answering tasks, such as, visual question answering, visual entailment, visual reasoning. In contrast to previous works using contrastive or generative captioning training, we propose a novel and simple recipe to pretrain a vision-language joint model, which is multi-task as well, and uses the entire architecture end-to-end. Our results, which are in the challenging open-vocabulary generative setting, show state-of-the-art performance, zero-shot generalization, robustness to forgetting.

1. Multi-Task Learning for VQA

Visual Question Answering (VQA) is a challenging task as it involves deeper understanding of both visual and language inputs. For intelligent VQA systems it is desirable that they operate with natural questions and answers and are able to generalize to other tasks, not seen during training. Commonly used fine-tuned models [3] tend to exhibit larger rates of catastrophic forgetting on new tasks [6].

We propose ‘Answer-Me’ which unifies visual question answering tasks and aims to answer a variety of natural language questions towards an image (fig:motivation). The gist of the method is multi-task, task-aware training, which is able to respond according to the question’s intent. This is combined with a novel pretraining which trains the entire encoder-decoder vision-language model simultaneously using only noisy data, and is also multi-task itself. This allows for natural language questions and free-form (open-vocabulary) outputs to answer accordingly, without additional prompts. Answer-Me generally outperforms multi-task SOTA methods, despite working in the challenging open-vocabulary setting, generalizing well to novel tasks.

Main architecture. Our model consists of an image encoder (ResNet) and text encoder and a Transformer fusion module. Our experiments are based on a ResNet-50 and T5-base model, and we scale it 3x by using ResNet-101 and T5-large. The image and language features are provided to a fusion module. The output of the fusion module is used as input to the text decoder, which produces free-form text for all Answer-Me tasks. While existing works have proposed similar fusion methods, ([3, 4]), the pretraining method and generalisation abilities without forgetting are new, using only raw images and no region proposals.

1.1. Pretraining for multi-task learning

In order to enable the model to address new tasks, i.e., to respond to unseen question types and answer adequately, we take advantage of a unique pretraining designed to train all the components of a model. Unlike previous work, this pretraining strategy is targeted towards training the entire encoder-decoder model, it exercises various pathways in the model, which makes it suitable for various question-answering tasks. Another key advantage of this approach is that the training loss (cross entropy over the tokens) is simple and shared for all tasks. Specifically, for each sample, we have an (image, text) pair, obtained from the pretraining data. To train all parts of the model, we design a mix of four tasks: (1) image captioning. Here the input text is ‘caption the image’ and the target text is the caption. This task mostly trains the text decoder and fusion layers. (2) caption completion. Here the input is 10–40% of the caption text and the target text is the remaining caption. This encourages training of the entire model. (3) text MLM [5]. Here the input is the caption with 25% of the words masked out, the target text is the missing words. This trains the en-
tire model. (4) **Image text matching** [3]. Here the input is either the image caption or a random caption, the target text is ‘true’ or ‘false’ if the caption matches the image or not.

2. Experiments

**Multi-task training.** The multi-task training is done by taking a set of $N$ tasks and mixing them together, sampled so that a batch consists of an equal amount of each dataset, i.e., batch size/N samples from each task. Since we use a text generation setting for the tasks, the loss is computed over the tokens, all using the same vocabulary. We use the T5 vocabulary with 32K tokens, for all experiments.

**Datasets.** We use the following datasets to address a number of VQA tasks: VQA2.0 [1], Visual Entailment (SNLI-VE) [14], Natural Language for Visual Reasoning (NLVR2) [13], GQA [9], and VizWiz [7] which is a VQA dataset collected by visually impaired users. The CC12m [2] is used for pretraining only.

**Evaluation.** We follow the evaluation protocols established in prior work, and use standard adopted metrics to measure performance. However, instead of training a classification output layer, we use a large, open vocabulary and generate text answers.

**Experimental results.** In tab:sota we compare to the state-of-the-art (SOTA) multi-task methods, such as UniT [8], 12-in-1 [11], GPV [6]. Our model generally out-performs others, despite using open-vocabulary.

We then test the capabilities of the Answer-Me models and their potential for skill transfer. I.e., we compare how a model performs when a task is included in the training mix vs. a task outside the mix. tab:main compares Answer-Me trained on single tasks vs. different task mixtures in both standard and Zero-Shot (ZS) evaluation. We observe that the mixtures provide competitive results, outperforming or on par with single fine-tuned (FT) models, while using a single model. As seen, more tasks in the mixture improve the performance across all datasets, so does scaling. Importantly, the same conclusions are observed for ZS, where multi-task is able to improve performance and reduce the gap to supervised training (we make sure there is no ‘leakage’ to the test set for each experiment). We note how VizWiz has very low ZS results, as it is very challenging.

**Answer-Me prevents catastrophic forgetting** While pretraining and fine-tuning, as is customarily done in pre-

---

**Table 1.** Comparison of Answer-Me (8 tasks) to SOTA multi-task models.

<table>
<thead>
<tr>
<th>Approach</th>
<th>VQA(dev)</th>
<th>NLVR2</th>
<th>SNLI-VE</th>
<th>GQA</th>
<th>VizWiz</th>
</tr>
</thead>
<tbody>
<tr>
<td>VL-BART [4]</td>
<td>71.3</td>
<td>70.3</td>
<td></td>
<td>60.5</td>
<td></td>
</tr>
<tr>
<td>VL-TS [4]</td>
<td>70.3</td>
<td>73.6</td>
<td></td>
<td>60.8</td>
<td></td>
</tr>
<tr>
<td>12-in-1 [11]</td>
<td>72.57</td>
<td>78.44</td>
<td>76.78</td>
<td>60.12</td>
<td>-</td>
</tr>
<tr>
<td>UniT (Coco init.) [8]</td>
<td>66.97</td>
<td>73.16</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Table 2.** Experiments comparing Answer-Me multi-task learning.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Model</th>
<th>VQA2.0</th>
<th>NLVR2</th>
<th>SNLI-VE</th>
<th>GQA</th>
<th>VizWiz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer-Me, PT, ZS</td>
<td>Single</td>
<td>25.3</td>
<td>32.5</td>
<td>22.7</td>
<td>40.9</td>
<td>2.3</td>
</tr>
<tr>
<td>Answer-Me 4 tasks, ZS</td>
<td>Single</td>
<td>30.0</td>
<td>42.5</td>
<td>34.1</td>
<td>42.3</td>
<td>9.7</td>
</tr>
<tr>
<td>Answer-Me 8 tasks, ZS</td>
<td>Single</td>
<td>35.0</td>
<td>44.7</td>
<td>37.3</td>
<td>44.2</td>
<td>10.3</td>
</tr>
<tr>
<td>Answer-Me 8 tasks, 3x</td>
<td>Single</td>
<td>39.2</td>
<td>48.3</td>
<td>41.1</td>
<td>47.2</td>
<td>11.4</td>
</tr>
<tr>
<td>Answer-Me, PT, 4 tasks</td>
<td>Single</td>
<td>64.8</td>
<td>71.5</td>
<td>77.2</td>
<td>72.1</td>
<td>71.5</td>
</tr>
<tr>
<td>Answer-Me, PT, 8 tasks</td>
<td>Single</td>
<td>65.1</td>
<td>71.7</td>
<td>77.5</td>
<td>72.8</td>
<td>72.4</td>
</tr>
<tr>
<td>Answer-Me, PT, 8 tasks, 3x</td>
<td>Single</td>
<td>73.6</td>
<td>73.9</td>
<td>85.8</td>
<td>77.5</td>
<td>75.3</td>
</tr>
</tbody>
</table>

---

**Table 3.** Fine-tuning vs multi-task Answer-Me. Fine tuned models tend to forget (first/second rows), even if original mix shows good within-data and out-of-sample (Zero-Shot) generalization (first rows). Additional fine-tuning seems to recover the losses within a task (first/third rows), but costs $N$ times the cost in training, and performance on the other task deteriorates again. Interestingly, this model performs even worse than the original out-of-sample mixture on the second task. Training on many tasks in the mix maintains performance for both tasks (last row).

---

**References**


