CAPWAP: Captioning with a Purpose

Adam Fisch\textsuperscript{1}  Kenton Lee\textsuperscript{2}  Ming-Wei Chang\textsuperscript{2}  Jonathan H. Clark\textsuperscript{2}  Regina Barzilay\textsuperscript{1}
\textsuperscript{1}Massachusetts Institute of Technology, \textsuperscript{2}Google Research
\{fisch, regina\}@csail.mit.edu, 
\{kentonl, mingweichang, jhclark\}@google.com

Abstract

The traditional image captioning task uses generic reference captions to provide textual information about images. Different user populations, however, will care about different visual aspects of images. In this paper, we propose a new task, Captioning with A Purpose (CAPWAP). Our goal is to develop systems that can be tailored to be useful for the information needs of an intended population, rather than merely provide generic information about an image. In this task, we use question-answer (QA) pairs—a natural expression of information need—from users, instead of reference captions, for both training and post-inference evaluation. We show that it is possible to use reinforcement learning to directly optimize for the intended information need, by rewarding outputs that allow a question answering model to provide correct answers to sampled user questions. We convert several visual question answering datasets into CAPWAP datasets, and demonstrate that under a variety of scenarios our purposeful captioning system learns to anticipate and fulfill specific information needs better than its generic counterparts, as measured by QA performance on user questions from unseen images, when using the caption alone as context.

1 Introduction

The image captioning task typically selects for captions having high similarity with generic human references. While this task definition has driven much of the research in the field, the end-purpose of these captions is not always clearly articulated. We argue that (1) generic annotations may not be representative of users’ information needs, (2) user questions are a more natural way of articulating information needs, and (3) optimizing captions to provide correct answers to those questions allows training to focus on information need. For example, in the VizWiz mobile application (Bigham et al., 2010), visually impaired users upload images from their everyday lives, along with questions about them that need to be answered. These questions serve as a powerful signal for aspects of the image that they find important.

Consider the image in Figure 1, where an annotator might provide a generic caption such as There is a green bus. This may be used to answer: What color is the bus? However, it would provide no utility to a user asking: Where is this bus headed? In fact, examples from VizWiz demonstrate a clear disconnect between the type of information provided by

<table>
<thead>
<tr>
<th>Task</th>
<th>Caption</th>
<th>Information Need</th>
</tr>
</thead>
<tbody>
<tr>
<td>Captioning</td>
<td>There is a green bus.</td>
<td>(Unspecified)</td>
</tr>
<tr>
<td>Visual QA</td>
<td>(Unspecified)</td>
<td>Where’s it headed?</td>
</tr>
<tr>
<td>CAPWAP</td>
<td>At least three people are boarding the #14 bus to Bembridge.</td>
<td>Which bus is this? Where’s it headed? How many people are boarding?</td>
</tr>
</tbody>
</table>

Figure 1: The informational purpose of generic captioning is not clearly defined, and VQA provides only reactionary information. The objective of the CAPWAP task is ultimately to provide more informative captions that specifically anticipate and satisfy users’ potential needs. In CAPWAP, we use QA as an implicit signal for information need: e.g., in the image above, a good caption that has been generated in advance should be able to be used to answer, Where is this bus headed?

*Work primarily completed while interning at Google.
Table 1: The CAPWAP task combines elements of generic image captioning with visual question answering. Training consists of images paired with visual questions and answers. A CAPWAP model should directly anticipate user information needs by outputting captions that can be used to answer future questions drawn from a distribution similar to the training data. Accordingly, the “QA” function represents the inference of an answer to a question using the generated caption as context. We approximate this with a strong automatic question answering model.

Today’s systems (e.g., arbitrary descriptions of entities and actions) versus what visually-impaired users need to know (e.g., fine-grained details to help make decisions).

Here, we propose an alternative framing for captioning: Captioning with A Purpose (CAPWAP). We do not assume the existence of a universal caption distribution. A good caption is highly subjective; different users will care about different aspects of a given image. Instead, we assume a distribution of visual question-answer pairs that are representative of population’s information needs. Here we aim to map images to text that can serve as context to answer likely questions under this distribution. At test time, the goal is to anticipate similar user questions for a new image, and implicitly answer them before they even need to be asked.

We use image-question-answer triplets as supervision, and require the model to generate from the latent space of captions that provide contextual support for the answer (Table 1). Within our task definition, any sampled caption that can be used to answer these questions is considered useful. Under this formulation, very different captions may be scored identically if they deliver the same content—regardless of word choice. Note that this is different from either standard visual question answering (VQA) or query-focused summarization: the target questions are not available prior to generation; at test time, they are used only for evaluation.

Existing approaches cannot be readily applied in this setting, as there are no gold reference captions for training—and off-the-shelf captioning systems transfer quite poorly (§6). To address the new learning challenge that arises in CAPWAP, we propose a novel model-in-the-loop reinforcement learning (RL) approach that acts as a strong baseline for this task. Our approach assumes a fixed question answering (QA) system that predicts an answer to a question using some input context. The captioning model receives a reward if it generates text which the QA system can use to predict the correct answer. Applying RL, however, is nontrivial. A naïve exploration of the caption generation space can lead to sparse rewards—resulting in long training times and disappointing quality. We show that our approach can be significantly improved by using a novel, synthetic pre-training routine to push the initial policy towards areas of high-reward.

We repurpose four VQA datasets for CAPWAP: VQA (Goyal et al., 2017), GQA (Hudson and Manning, 2019), Visual7W (Zhu et al., 2016), and VizWiz (Gurari et al., 2018). These datasets range in style from synthetic QA pairs (GQA) to natural information-seeking questions asked by visually-impaired users (VizWiz). We find that our method produces significantly more informative captions with respect to the given questions (up to $3.8 \times$ exact match), compared to models trained on generic captions from COCO (Lin et al., 2014).

Our key contributions are as follows:

1. We define a new task (CAPWAP) that generates image captions for the purpose of fulfilling specific information needs expressed by different target user populations.

2. We demonstrate that our information-need-driven model can generate much higher quality captions on this task than those of state-of-the-art traditional generic captioning systems.

3. We propose a novel synthetic pre-training routine that greatly improves the performance of reinforcement learning under this new paradigm.
2 Related Work

Since the early days of the field, human-written references have been used for the supervised training and evaluation of text generation systems, including image captioning, summarization, and other related applications (Edmundson, 1969; Lin and Hovy, 2003; Ordonez et al., 2011; Vinyals et al., 2015). Recently, researchers have begun to consider a multitude of different objectives for reference comparison (Böhm et al., 2019; Gao et al., 2019), or even parametric regressions trained on human judgements (Louis and Nenkova, 2013; Peyrard and Gurevych, 2018). Though diverse in approach, each ultimately relies on designing a robust general-purpose metric. In practice, engineering such a metric is challenging—if at all possible (Spärck Jones, 1994, 1999). Here we take a more empirical approach by relying on the information need expressed by users’ questions.

Many studies have observed that reference-trained captioning models suffer from systematic usability issues—including being rigid, neglecting relevant image aspects, and regurgitating frequent phrases (Wang et al., 2017; Dai et al., 2017). As a result, much effort has been focused on developing secondary, corrective objectives—for instance, “discriminability” losses encouraging captions to be unique (Dai and Lin, 2017; Liu et al., 2018; Luo et al., 2018). While these measures provide some fixes, they do not necessarily reflect user information needs—a central concept in CAPWAP.

The idea of using QA for assessing information quality has been proposed in recent work for text summarization (Arumae and Liu, 2019; Eyal et al., 2019; Scialom et al., 2019). The primary distinctions with our work are both the domain (images) and how questions are obtained—both of which impact the task objective and learning procedure. In this prior work, questions are generated programatically (e.g., following Hermann et al., 2015). Such “questions” may not necessarily reflect real user preferences. Our work focuses on QA not as just another method to improve standard reference-based metrics, but as a key, flexible way of formulating user information need—and as such we focus on challenging, real QA datasets. Furthermore, we train on this signal, rather than rely on it solely for evaluation (Wang et al., 2020).

Efforts to leverage VQA resources to drive image captioning, and vice-versa, via variations of transfer learning, have also received extensive interest in recent years (Li et al., 2018; Wu et al., 2019; Yang and Xu, 2019). As opposed to optimizing metrics for specific VQA or supervised captioning benchmarks, the primary focus in CAPWAP is on modeling the target user population in order to anticipate the correct information-need.

In a similar vein, VQA and textual QA resources have also been leveraged for active learning (Shen et al., 2019; Li et al., 2017), where the model learns to query its environment for information it is uncertain about to help improve its performance on the given task. The key distinction with our work is the directionality of the questions. In CAPWAP, the model uses questions posed by the users to infer their latent information need—which is a distinctly different, and quite challenging, setting.

3 Problem Formulation

We begin by formulating the CAPWAP task. In our setting, questions and answers are the only source of direct supervision assumed during training. At test time, the model is not given questions in advance, but rather must anticipate the information need of the user, and generate captions that answer the forthcoming questions in expectation.

Task Setting: Given an image $x$ the model must output a caption $y$, such that $y$ entails the answer $a$ for a question-answer pair $(q, a)$ sampled from some underlying distribution $D$. Examples from $D$ are given during training, but are not known in advance by the generation model at test time.

Information Need: We assume that the QA data from $D$ is derived by the following process:

1. an image $x$ is drawn from distribution $p(x)$;
2. a question-answer pair $(q, a)$ targeting an informative detail of $x$ perceived as important to a user in $D$ is drawn from distribution $p(q, a|x)$.

The operating assumption is that the marginal distribution over $(q, a)$ pairs represents the visual interests of the typical user. In other words, answers to common questions represent the type of information that is often considered important. This is comparable to content selection (Peyrard, 2019).

Question Anticipation: We do not assume the existence of a “gold” caption. Rather, the caption $y$ is assumed to be a latent variable, and $G_{\theta}(y|x)$ is a
We estimate this using the accuracy of a pre-trained VQA model $M(q, y)$, when using $y$ as context for $q$. CAPWAP requires maximizing the expectation:

$$\arg\max_\theta E_{G_\theta(y|x)} \left[ E_{P(a|x)} \left[ R(y, q, a) \right] \right]$$

where $\theta$ parameterizes $G_\theta(y|x)$, and we choose our reward to be $R(y, q, a)$, any appropriate accuracy metric for comparing the output of $M(q, y)$ with $a$ (expressed as ACCURACY in Table 1).

**CAPWAP vs. Other Tasks:** Table 1 compares our setting to those of both standard (generic) captioning and visual question answering. Both standard captioning and CAPWAP models output a single caption per image, but CAPWAP does not compare to references. Both VQA and CAPWAP models are trained and evaluated with QA data, but CAPWAP does not provide the question prior to generation. VQA models output single answers, whereas CAPWAP models output anticipatory contexts.

4 **An Approach to CAPWAP**

Given that we only have access to question-answer pairs during training, but not during inference, how can we learn a model for this task? Eq. 1 naturally lends itself to a reinforcement learning (RL) framework where the model receives a reward $r$ (e.g., $r = R(y, q, a)$) for each generated caption $y$ and training QA pair $(q, a)$. $G_\theta(y|x)$ can be cast as a policy, and updated with policy gradients.

Optimizing such a policy, however, poses a technical challenge because the model is only rewarded for correct (or partially correct) answers, which is initially a rare event. Transferring $G_\theta(y|x)$ from generic captioning data can be a useful starting point. Our method then follows this recipe:

1. Initialize $G_\theta(y|x)$ using fully-supervised off-the-shelf captioning data, $(\tilde{x}, \tilde{y}) \sim D_{\text{generic}}$.
2. Fine-tune $G_\theta(y|x)$ using policy gradient on targeted visual QA data, $(x, q, a) \sim D_{\text{target}}$.

In Sections 4.1 and 4.2 we detail our model for $G_\theta(y|x)$, and the above training procedure.

Note that $D_{\text{generic}}$ is assumed to be out-of-domain for our intended captioning purpose, $D_{\text{target}}$. Since we are interested in diverse user-generated questions and information needs, the generic captioning data can often diverge dramatically from our end goal. To improve transfer, in Section 4.3 we further develop a novel mechanism for automatically generating in-domain synthetic data that can be used as pre-training for guiding $G_\theta(y|x)$ towards balanced areas of high reward in $D_{\text{target}}$.

4.1 **Model Architecture**

We briefly describe our base captioning model, which consists of a Faster R-CNN and Transformer-based encoder-decoder, following the sequence-to-sequence framework common in state-of-the-art image captioning systems (Anderson et al., 2018; Vinyals et al., 2015; Zhou et al., 2019). See Appendix A for full technical details. Given an image $x$, we first represent it as a sequence of detected object bounding box embeddings, computed from a pre-trained Faster R-CNN model (Anderson et al., 2018). We then generate caption word-pieces $y = (y_1, \ldots, y_n)$ using a Transformer-based architecture (Vaswani et al., 2017).

4.2 **Policy Training**

We describe our RL framework for training our captioning model using QA data. See Appendix B for additional technical details, including hyper-parameter settings and optimization choices.
Initialization: We initialize $G_\theta(y|x)$ using maximum likelihood estimation (MLE) on a corpus of out-of-domain generic captions $(\hat{x}, \hat{y})$, as common practice (Ranzato et al., 2016). This warm-starts our policy with an initial set of grounded image concepts, albeit not necessarily the ones we ultimately care about. Given the generic reference $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_n)$, we minimize the cross-entropy:

$$L_{XE}(\theta) = -\sum_{i=1}^{n} \log G_\theta(\hat{y}_i | \hat{x}, \hat{y}_{j<i})$$  \hspace{1cm} (2)

QA Model: We implement the QA model $M$ using a BERT$_{LARGE}$ extractive model fine-tuned on SQuAD 2.0 (Rajpurkar et al., 2018)—which contains unanswerable questions. As an extractive model, $M$ predicts a span $y_{i...j}$. Important for our use-case, $M$ is both able to be accurate when predicting the answer $a$ when $a$ is present in $y$, and also able to abstain from answering when $a$ is not logically entailed (i.e., predict “no answer”).

QA Reward: We take $R(y, q, a)$ from Eq. 1 as the F1 score of the predicted answer with the gold answer. We control for reward noise with a confidence threshold for predicting “no answer.”

Policy Gradient: We use REINFORCE with a baseline (Williams, 1992) to compute the policy gradient $\nabla_\theta L_{QA}(\theta)$ of the QA reward:

$$-E_{y|x}[(R(y, q, a) - b) \nabla_\theta \log G_\theta(y|x)]$$ \hspace{1cm} (3)

We take $b$ as $R(\hat{y}, q, a)$, where $\hat{y}$ is the argmax (test-time prediction) of $G_\theta$, following the self-critical method of Renne et al. (2017).

4.3 Synthetic Policy Pre-Training

In the beginning of training, the generated captions typically do not correctly answer many questions, leading to almost no reward signal. More formally, the reward is sparse if the policy $G_\theta(y|x)$ is not well-initialized. As a result, REINFORCE becomes extremely sample-inefficient. When the target distribution is strikingly divergent from the one present in the generic captioning data—a key setting in this work—supervised pre-training on the out-of-domain data does not yield a usable initialization. As a substitute, we derive a method for generating a synthetic dataset of captions $D_{\text{synthetic}}$ with high-reward as a form of guided policy search (Levine and Koltun, 2013). The full method then consists of three stages that train on the three datasets: $D_{\text{generic}} \rightarrow D_{\text{synthetic}} \rightarrow D_{\text{target}}$.

For the extractive QA model to possibly yield a positive reward, the answer must be a span of the caption. When the question and answer are known in advance, it is typically fairly simple to reverse engineer a candidate caption that meets this constraint (e.g., by inverting $wh$-movement). Figure 3 demonstrates this concept. If we have an auxiliary model $F_\phi(y|x, q, a)$ that can automate this reverse engineering step, we can synthetically generate captions to use for pre-training, as in Eq. 2.\footnote{Methods for constrained decoding (Anderson et al., 2016; Hokamp and Liu, 2017, inter alia) that enforce $a \subseteq y$ are related, yet complementary, and can be incorporated into any $F_\phi$. It is more important to ensure that not only is the answer contained in the caption, but also that it is logically supported.}

QA Conditional Model: Motivated by this, we learn $F_\phi(y|x, q, a)$ by explicitly conditioning on QA pairs when generating a caption that supports the answer span by design. Concretely, we include the word-pieces of the question $q = (q_1, \ldots, q_l)$ and answer $a = (a_1, \ldots, a_m)$ as inputs when decoding $y$, while $y$ satisfies $M(y, q) = a$.

How do we train $F_\phi(y|x, q, a)$ effectively without access to any paired data $(x, q, a, y)$? We create automatic $(\hat{x}, \hat{q}, \hat{a}, \hat{y})$ examples from the out-of-domain generic captioning data used in Section 4.2 by using the (text-based) question generation model $M^{-1}$ of Alberiti et al. (2019). At a high level, given a generic caption $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_n)$, this inverse model (1) picks an answer span $\hat{a} \subseteq \hat{y}$, (2) generates a question $\hat{q}$ following some inferred distribution $p(q|\hat{a}, \hat{y})$, and (3) confirms that the sample obeys “round-trip filtering”, i.e., that the original QA model answers the synthetic example correctly $(M(\hat{q}, \hat{y}) = \hat{a})$. We then train the model for $F_\phi$ using conditional cross-entropy:

$$L_{CXE}(\phi) = -\sum_{i=1}^{n} \log F_\phi(\hat{y}_i | \hat{x}, \hat{q}, \hat{a}, \hat{y}_{j<i})$$ \hspace{1cm} (4)

Synthetic Data Generation: After training, we transfer $F_\phi(y|x, q, a)$ with fixed weights to generate reverse engineered captions $\hat{y}$ using the
true \((x, q, a, \text{null})\) examples from our target QA datasets. For each example, we decode the top-\(k\) captions using beam search, and keep those with \(R(\hat{y}, q, a) \geq c\), where \(c\) is a threshold (e.g., \(c = 1.0\) for an exact match F1 score). These examples paired with the high-scoring captions are used to create the synthetic captions dataset \(D_{\text{synthetic}}\). We then use \(D_{\text{synthetic}}\) as further weak supervision for initializing \(G_{\theta}(y | x)\), again following Eq. 2. See Appendix C for additional technical details.

5 Experimental Setup

Evaluation: Our primary evaluation assumes a dataset of questions and answers about images. Conceptually, if the correct answers are supported by the generated caption in expectation, then we consider it to be sufficiently informative.\(^2\)

Automatic Evaluation: Our automatic proxy of informativeness utilizes the state-of-the-art extractive question answering model \(M\) described in Section 4.2 that is trained on SQuAD 2.0.\(^3\) \(M\) is applied to the given QA pair, with the generated caption as the “context.” We report EM, measuring exact match with the gold answer, and F1, measuring word overlap. If there are multiple answers, then we take the maximum score over all.

Human Evaluation: For human evaluation we ask raters to judge whether a caption is less, equally, or more informative than another caption with respect to the question-answer pair. We also gather human ratings for two properties that are desirable regardless of the target audience: (1) fluency (whether the caption is grammatical and coherent) and (2) fidelity (whether the caption makes any false assertions regarding what is in the image).

5.1 Datasets

We evaluate our method on four converted visual question answering datasets. We filter questions that are unanswerable, or have “yes/no” or non-alphabetic answers.\(^4\) Appendix D gives additional size, splitting, and pre-processing details.

COCO (Lin et al., 2014): For all experiments, we use COCO as the source of out-of-domain generic captions for pre-training. COCO contains images covering 80 object categories and various scenes gathered from Flickr, paired with five human-written reference captions.

CapVQA (Goyal et al., 2017): VQA v2.0 originally contains questions written by crowd-workers where the prompt was to write queries that are easy for humans to answer, but challenging for a hypothetical robot that mainly knows only about objects. VQA is the only dataset we consider that fully covers the same images as COCO.

CapGQA (Hudson and Manning, 2019): GQA contains challenging compositional questions derived from scene graphs of everyday images using various human-specified grammars.

CapVisual7W (Zhu et al., 2016): Visual7W contains questions written by crowd-workers about objects that, in general, require richer and longer answers than those in VQA. We use only the “telling” split of the dataset (i.e., the questions that require open-ended natural language answers).

CapVizWiz (Gurari et al., 2018): VizWiz consists of natural visual questions asked by visually-impaired users of a mobile application who were seeking answers to their daily visual needs. Each question is answered by a remote assistant.

5.2 Generic Captioning Models

In addition to our baseline captioning model trained only to maximize the likelihood of COCO references (MLE in the tables), we compare to two state-of-the-art generic image captioning methods (also trained on COCO data). Huang et al. (2019) directly optimizes the CIDEr metric with policy gradients, while Luo et al. (2018) optimizes both CIDEr and a “discrimination” loss intended to encourage models to describe each image’s uniquely identifying aspects. These models are included in order to highlight the differences in applicability between off-the-shelf models trained for generic image captioning versus those for CAPWAP.

6 Results

In the following, we address several key research questions relating to our approach to CAPWAP, and the broader assumptions, strengths, and limitations of using QA to drive the process.
Table 2: Does the proposed approach better fulfill information needs? We show question answering test performance when applying an exhaustive question answering model on predicted captions (see Table 1). Existing captioning models trained on generic references (rows 2-4)—or even the generic references themselves (row 1)—do not capture the information requested by different QA datasets. Applying our RL method for tailoring towards CapWAP (row 5) leads to more informative captions with respect to those questions (and by extension, for the assumed end-users). Adding synthetic pre-training data (+ SYN) improves results on several datasets (row 6).

<table>
<thead>
<tr>
<th>Model</th>
<th>CapVQA</th>
<th>CapGQA</th>
<th>CapVisual7W</th>
<th>CapVizWiz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human reference (from COCO)</td>
<td>16.5</td>
<td>25.7</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Luo et al. (2018)</td>
<td>12.0</td>
<td>20.1</td>
<td>9.6</td>
<td>13.9</td>
</tr>
<tr>
<td>Huang et al. (2019)</td>
<td>16.0</td>
<td>25.0</td>
<td>9.9</td>
<td>14.9</td>
</tr>
<tr>
<td>Our generic baseline: MLE</td>
<td>16.8</td>
<td>25.2</td>
<td>8.0</td>
<td>11.1</td>
</tr>
<tr>
<td><strong>Our CapWAP model: RL</strong></td>
<td>23.1</td>
<td>32.3</td>
<td>15.7</td>
<td>19.3</td>
</tr>
<tr>
<td><strong>Our CapWAP model: RL + SYN</strong></td>
<td>24.2</td>
<td>33.2</td>
<td>16.6</td>
<td>19.8</td>
</tr>
</tbody>
</table>

Table 3: Do raters think that the proposed approach provides more informative captions? Human evaluation of the informativeness of captions with respect to our QA datasets agrees with our automatic evaluation—finding our model to have better information coverage than MLE, our baseline without QA rewards.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Informativeness</th>
</tr>
</thead>
<tbody>
<tr>
<td>A &gt; B</td>
<td>27% 20%</td>
</tr>
<tr>
<td>B &gt; A</td>
<td>31% 20%</td>
</tr>
<tr>
<td>CapVQA</td>
<td>38% 22%</td>
</tr>
<tr>
<td>CapGQA</td>
<td>37% 20%</td>
</tr>
<tr>
<td>CapVisual7W</td>
<td>27% 20%</td>
</tr>
<tr>
<td>CapVizWiz</td>
<td>38% 22%</td>
</tr>
</tbody>
</table>

Table 4: Does the proposed approach tailor to specific information need? We show transfer performance (dev F1) of the RL + SYN policies learned on different QA datasets. The in-domain F1 peaks indicate that the model is producing distribution-specific captions.

<table>
<thead>
<tr>
<th>Train</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CapVQA</td>
<td>33.8</td>
</tr>
<tr>
<td>CapGQA</td>
<td>24.8</td>
</tr>
<tr>
<td>CapVizWiz</td>
<td>26.0</td>
</tr>
<tr>
<td>CapVisual7W</td>
<td>23.9</td>
</tr>
<tr>
<td>Generic</td>
<td>25.5</td>
</tr>
</tbody>
</table>

Evaluation of Generic Captions: We begin by empirically verifying our introductory claim that training on generic reference captions can poorly reflect the varying, user-specific information need. Table 2 presents the results of the baseline generic captioning systems when evaluated in terms of how well the predicted captions support QA over different distributions. Though they are strong methods as measured on the COCO benchmark, unsurprisingly, they still fail to capture all the information necessary to answer diverse visual questions. Performance on CapVizWiz is exceptionally poor; the visually-impaired users ask for information strikingly different than what is represented in COCO. The causes of this poor performance go beyond simple limitations in the current state-of-the-art models: the target references themselves are insufficient. For example, on CapVQA, where the images overlap with COCO and thus human captions are directly available, the average performance of these “gold” references is only slightly better—supporting our conjecture that good captions for one purpose are not necessarily good for another, even on the same images.

Adaptation to Information Need: We next test the effectiveness of our proposed approach at tailoring captions to meet the specific information need stipulated by our datasets. Our results in Table 2 demonstrate significant improvements by our QA-driven models (RL and RL + SYN) across all four datasets—achieving an average gain of 8.0 absolute F1. Notably, we improve by 7.5 EM over the average human caption on CapVQA, and by 16.5 EM over the best generic model on CapVizWiz. Table 4 further illustrates that the adaptation process is indeed tailored to the respective QA datasets. The improvements on our automatic QA-based metrics (using the proxy model \( M \)) also translate to human judgements. Table 3 presents the results of our human \( A/B \) test of our proposed model vs. the MLE baseline. Relative to MLE, we find that our method is significantly more informative with respect to unseen QA pairs across all datasets. As expected, the largest improvements are on the datasets whose questions deviate significantly from the generic COCO content (e.g., CapVizWiz).
Figure 4: Example outputs comparing our model trained for CAPWAP (RL + SYN) with the baseline model trained on generic COCO references (MLE). These examples are representative of the way in which the various datasets ask about image content. Tendencies include colors for CapVQA, spatial relations for CapGQA, higher-level concepts in CapVisual7W, and OCR in CapVizWiz. Note that our approach to CAPWAP does not (and likely cannot ever perfectly) anticipate *all* unseen questions—but is distributionally closer in terms of content selection.

A = RL + SYN vs. B = RL.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Fluency</th>
<th>Fidelity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$A &gt; B$</td>
<td>$B &gt; A$</td>
</tr>
<tr>
<td>CapVQA</td>
<td>57%</td>
<td>26%</td>
</tr>
<tr>
<td>CapGQA</td>
<td>84%</td>
<td>6%</td>
</tr>
<tr>
<td>CapVisual7W</td>
<td>69%</td>
<td>19%</td>
</tr>
<tr>
<td>CapVizWiz</td>
<td>38%</td>
<td>37%</td>
</tr>
</tbody>
</table>

Table 5: Does synthetic data improve secondary measures of caption quality? Raters find that this strategy dramatically improves fluency and fidelity ($\S$5) when compared to a model with only on-policy sampling.

Importance of Synthetic Pre-training: A deficiency of QA-based rewards is that they neither explicitly enforce text fluency, nor penalize the system when content is produced that is either not relevant or not true. On the other hand, when reference captions are available, it is easy to learn a fluent language model. Table 5 shows that incorporating synthetic, guiding “silver” samples from our auxiliary QA conditional model $F_\phi(y|x, q, a)$ ($\S$4.3) to bridge the gap between $D_{\text{generic}}$ and each considered $D_{\text{target}}$ dramatically reduces the fluency and fidelity issues that arise from training solely with QA rewards. Ultimately, however, Table 6 shows that our model still suffers on these secondary metrics as compared to the reference-trained MLE baseline. This is a challenge that is shared with nearly all other comparable RL-based methods for text generation (e.g., Guo et al., 2018; Paulus et al., 2018, *et cetera*). Incorporating complementary fluency re-

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Fluency</th>
<th>Fidelity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$A &gt; B$</td>
<td>$B &gt; A$</td>
</tr>
<tr>
<td>CapVQA</td>
<td>37%</td>
<td>33%</td>
</tr>
<tr>
<td>CapGQA</td>
<td>30%</td>
<td>57%</td>
</tr>
<tr>
<td>CapVisual7W</td>
<td>37%</td>
<td>53%</td>
</tr>
<tr>
<td>CapVizWiz</td>
<td>34%</td>
<td>72%</td>
</tr>
</tbody>
</table>

Table 6: Do more informative captions come at a cost? Raters find that our tailored “purposeful” approach is less fluent more than half the time (left). Still, this system has greater or equal fidelity to the image content on most datasets (right). At a high level, while the system does give more relevant information, it may do so in a less fluent way, a direction we leave for future work.

Qualitative Discussion: The qualitative effect of our method is quite intuitive (see Figure 4 as well as Table 7). For example, many questions in CapGQA ask about spatial relations, which is reflected in the generated captions. On the other hand, CapVizWiz users often ask about detailed information about meals, and the adapted model attempts to provide a more useful description beyond a “plate of food.” Of the above datasets, note that only CapVizWiz consists of questions asked by genuinely interested users. Interestingly, this property unearths yet another challenge: CapVizWiz questions can be long-form and quite different from SQuAD, and reverse engineering them (using $F_\phi$) for pre-training is noisier (as evidenced by the performance of +SYN
Ablation Studies: Tables 7 and 8 show the effects calculated over the dev sets: average caption length (|y|), adjective production rate (adj%), and the total vocabulary size of the unique unigrams and bigrams emitted (|V^2|). Captions are measured in tokens (PTB-style), adjectives are identified using NLTK (Loper and Bird, 2002), and vocabulary size is measured in thousands. Both CAPWAP methods tend to produce longer captions, presumably with more descriptions (higher number of adjectives). Notably, RL + SYN manages to maintain more “natural” adjective production rates and richer language usage (in terms of bigram usage) than RL only, supporting the human quality ratings in Table 5.

<table>
<thead>
<tr>
<th>Model</th>
<th>CapVQA</th>
<th>CapGQA</th>
<th>CapVisual7W</th>
<th>CapVizWiz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>y</td>
<td>V^2</td>
<td></td>
</tr>
<tr>
<td>Generic: MLE</td>
<td>10.8</td>
<td>6.9</td>
<td>3.6</td>
<td>11.1</td>
</tr>
<tr>
<td>CAPWAP: RL</td>
<td>16.2</td>
<td>15.8</td>
<td>4.8</td>
<td>15.6</td>
</tr>
<tr>
<td>CAPWAP: RL + SYN</td>
<td>13.9</td>
<td>11.1</td>
<td>4.8</td>
<td>13.8</td>
</tr>
</tbody>
</table>

Table 7: How do the generated captions differ qualitatively? We present a number of automatic qualitative measures of caption content calculated over the dev sets: average caption length (|y|), adjective production rate (adj%), and the total vocabulary size of the unique unigrams and bigrams emitted (|V^2|). Captions are measured in tokens (PTB-style), adjectives are identified using NLTK (Loper and Bird, 2002), and vocabulary size is measured in thousands. Both CAPWAP methods tend to produce longer captions, presumably with more descriptions (higher number of adjectives). Notably, RL + SYN manages to maintain more “natural” adjective production rates and richer language usage (in terms of bigram usage) than RL only, supporting the human quality ratings in Table 5.

<table>
<thead>
<tr>
<th>Model</th>
<th>Reward</th>
<th>CapVQA</th>
<th>CapGQA</th>
<th>CapVisual7W</th>
<th>CapVizWiz</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RL + SYN</td>
<td>answer supported</td>
<td>42.5</td>
<td>24.7</td>
<td>33.8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w/o “no answer”</td>
<td>40.8</td>
<td>23.3</td>
<td>32.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>w/o QA model</td>
<td>46.4</td>
<td>22.4</td>
<td>31.5</td>
<td></td>
</tr>
<tr>
<td>Generic: MLE</td>
<td>None</td>
<td>32.9</td>
<td>17.1</td>
<td>25.5</td>
<td></td>
</tr>
</tbody>
</table>

Table 8: What is the impact of using the QA model to provide rewards? We present an ablation study across our different datasets when using the QA model with the “no answer” option or not, as well as a simple indicator reward, 1{a ⊆ y}, that simply measures if the answer string is present at all (without running the expensive QA model). Our results show that while the indicator reward increases the indicator metric (IN) the most, these are likely mostly spurious or disfluent generations. Using the QA model improves the F1 and EM scores across all datasets—and in all cases except one improves further when confidence is used.

in Table 5). While the artificial settings of the other datasets are not ideal, their diversity serves to demonstrate the flexibility of our approach.

Ablation Studies: Tables 7 and 8 show the effects of different design choices in our RL and RL + SYN models. A significant challenge for CAPWAP systems, as previously discussed and illustrated in Table 6, is learning information need while maintaining fluency. Table 7 shows how synthetic pre-training regularizes the model to stay closer to human-level production patterns. Similarly, Table 8 shows how using the QA model to provide rewards (as opposed to a simple keyword search) helps the model avoid spurious rewards.

Future Work: The CAPWAP paradigm introduces new challenges for learning effective systems, some of which our approach solves, and others which it still leaves open (e.g., maintaining fluency and fidelity). While some may be addressed by large-scale multi-modal models (Li et al., 2019; Tan and Bansal, 2019), it is still unclear whether they would fully cover the diversity of information that real users are interested in (e.g., OCR).

7 Conclusion

We defined and studied the CAPWAP task, where question-answer pairs provided by users are used as a source of supervision for learning their visual information needs. Our results indicate that measuring caption content by its ability to logically support the answers to typical QA pairs from a target audience is (1) not only feasible, but also (2) a good proxy for uncovering information need. We hope this work will motivate the image captioning field to learn to anticipate and provide for the information needs of specific user communities.

Acknowledgements

We thank the MIT NLP group, the Google AI Language team, and the anonymous EMNLP reviewers for their valuable feedback. AF is supported in part by an NSF Graduate Research Fellowship.

8763
References


Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In ACL.


Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. In NAACL.


Danna Gurari, Qing Li, Abigale Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey Bigham. 2018. VizWiz grand challenge: Answering visual questions from blind people. In CVPR.

Karl Moritz Hermann, Thomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In NeurIPS.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In ACL.

Lun Huang, Wenmin Wang, Jie Chen, and Xiao-Yong Wei. 2019. Attention on attention for image captioning. In ICCV.

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


Andrej Karpathy and Fei-Fei Li. 2015. Deep visual-semantic alignments for generating image descriptions. In CVPR.


Jiwei Li, Alexander H. Miller, S. Chopra, Marc’Aurelio Ranzato, and Jason Weston. 2017. Learning through dialogue interactions by asking questions. In ICLR.


Qing Li, Jianlong Fu, Dongfei Yu, Tao Mei, and Jiebo Luo. 2018. Tell-and-answer: Towards explainable vi-
sual question answering using attributes and captions. In *EMNLP*.


Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *ACL*.


We then encode \( h \) where \( w \) (2018). We decode the caption auto-
Transformer sequence-to-sequence model. As the
The captioning architecture we use is a standard
vision-language pre-training for image captioning and
Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-
Fei, 2016. Visual7W: Grounded question answering in
images. In CVPR.

A Model Architecture Details

The captioning architecture we use is a standard
Transformer sequence-to-sequence model. As the
model is not the main focus, we did not do any ex-
tensive hyper-parameter tuning or ablations beyond
ensuring that we had a reasonable baseline model
on COCO (114 CIDEr on COCO captions).

Image Encoder: For each image \( x \), we take
represent the region embeddings \( o_t \in \mathbb{R}^{2048} \)
of the bounding boxes for the \( k \) most confident ob-
ject detections. We use the pre-trained Faster R-
CNN (Ren et al., 2015) model of Anderson et al.
(2018). We then map each region embedding to
\( \tilde{o}_t \in \mathbb{R}^d \) using a single dense layer with a ReLU.
Inspired by the positional token embeddings in the
BERT model (Devlin et al., 2019), we then aug-
ment \( \tilde{o}_t \) with learned position (the rasterized co-
ordinate of the bounding box center), segment (a
constant “image” component identifier), and confi-
dence (the detection rank of the object) embeddings
to obtain the full object representation:
\[
\tilde{o}_t = \tilde{o}_t + p_t + s_t + c_t.
\]

Text Decoder: We decode the caption auto-
regressively—starting with the [CLS] token and
terminating on [SEP]. At each time-step \( t \) we
concatenate the image embeddings with special de-
limiters and the word-pieces decoded thus far, to
obtain a joint context:
\[
h = \{[IMG], \tilde{o}_1, \ldots, \tilde{o}_k, [CLS], w_1, \ldots, w_t\},
\]
where \( w_i \in \mathbb{R}^d \) is the word piece embedding (the
sum of token, position, and segment embeddings).
We then encode \( h \) using multi-layer Transformer,
and compute the probability of generating \( w_{t+1} \)
using a softmax over the 30,522 word-piece vocab-
ulary (the BERT vocabulary). For efficiency, we
encode whole sequences at a time with a left-to-
right attention mask: image regions may attend to
all other image regions, and tokens may attend to
all previous tokens and image regions.

Hyperparameters: In our experiments we use the
top 64 object regions and a 6-layer Transformer
with 512 hidden input units, 8 attention heads, and
2,048 hidden units in the intermediate feed-forward
layer. During inference we do beam search with a
beam size of 3 and a length penalty \( \alpha \) of 0.6
(Wu et al., 2016). We implemented our model in
Tensorflow (Abadi et al., 2015).

B Model Training Details

QA Model Threshold: During inference, the
QA model \( M(q, y) \) computes the probability of the
“no answer” option \( p_M(\text{NONE}|q, y) \) and
the probability of the most likely answer span
\( p_M(y_{i...j}|q, y) \). We adjust how precise this reward
is by treating the log odds ratio \( c \) of the “no an-
swer” vs. span options as a hyper-parameter when
choosing the prediction \( \hat{a} \):
\[
\log \left( \frac{p_M(y_{i...j}|q, y)}{p_M(\text{NONE}|q, y)} \right) \begin{cases} > c, & \hat{a} = y_{i...j} \\ \leq c, & \hat{a} = \text{NONE} \end{cases}
\]
Depending on the value of \( c \), we may only answer if
we are confident the answer is supported—not just
the most probable (e.g., based on answer-type)—
to avoid potentially spurious rewards obtained by
guessing or elimination. Table 8 shows an ablation
over some choices of \( c \).

Answer Normalization: For both training and
evaluation, we normalize the gold and predicted an-
swers by removing articles and punctuation when
comparing them (see Rajpurkar et al., 2016).

Policy Gradient: We approximate the policy
gradient (Eq. 3) using a single Monte-Carlo sample
\( y = (y_1, \ldots, y_n) \) from \( G(y|x) \). We accelerate
training by restricting samples to be from a set
of high-probability candidates with non-zero re-
ward (cf. Anderson et al., 2018; Narayan et al.,
2018, inter alia). We decode using beam search
and sample from the top-\( k \) beams (\( k = 16 \)).

Training: For MLE pre-training on \( D_{\text{generic}} \)
(Eq. 2), all examples are shuffled and divided into
mini-batches of 256 examples each. For RL adap-
tation to \( D_{\text{target}} \) (Eq. 3), we use a mini-batch size of
128. To help regularize the fluency of the model,
during RL training we continue to multi-task on the
supervised generic captions, as in MIXER (Ran-
zato et al., 2016). For both settings, we train for
a maximum of 120K steps and choose the best
model based on the dev set performance (using COCO CIDEr (Vedantam et al., 2015) for MLE pre-training and QA F1 for RL). For optimization, we use Adam (Kingma and Ba, 2015) with a linear warm-up and decay schedule. Training was performed on a 4 x 4 TPU, and took about 1-2 hours per experiment.

C Synthetic Pre-training Details

QA Conditional Model: We use the same basic architecture for \( F_\phi(y|x, q, a) \) as for the main captioning model \( G_\phi(y|x) \), and only introduce two new “question” segment and “answer” segment embeddings that we add to differentiate the conditional text from the generated text in the Transformer. The full input then becomes:

\[
h = \{[\text{IMG}], \hat{o}_1, \ldots, \hat{o}_k, [\text{Q}], q_1, \ldots, q_m, [\text{A}], a_1, \ldots, a_n, [\text{CLS}], w_1, \ldots, w_l\},
\]

where the segment delimiter, \( q_i \), and \( a_j \) vectors are defined the same way as—and shared with—the caption’s input word-piece embeddings \( w_i \) (see §A). We decode auto-regressively as before.

Training and Generating: We train \( F_\phi(y|x, q, a) \) using both the corpus of generic captions used for MLE pre-training in Section 4.2 (i.e., COCO) and additional Wikipedia text. We create automatic \( (x, q, a, y) \) and \( (null, q, a, y) \) examples for COCO and Wikipedia sentences, respectively. To offset biases present in the question generation model (which is out-of-domain for caption-styled text as it is trained on SQuAD), we add \( (x, null, a, y) \) examples from the generic captions by selecting random spans of \( y \), using the sampler of Joshi et al. (2019) (random spans with Poisson distributed lengths). Algorithm 1 illustrates the full synthetic data generation process.

Algorithm 1 Synthetic data generation procedure for policy pre-training.

Definitions: \( D_{\text{generic}} \) is assumed out-of-domain generic captioning data with input images \( x \) and supervised reference captions \( y \). \( D_{\text{target}} \) is the in-domain target QA data with input images \( x \), questions \( q \), and answers \( a \). \( M \) is the automatic QA model used in this paper for evaluation (Table 1). \( M^{-1}(y) \) is a pre-trained QA generation model, that takes in some context \( y \) and outputs a predicted QA pair \( (q, a) \).

1: function TRAIN\((D_{\text{generic}}, M^{-1}, T)\)
2: \( \phi \leftarrow \text{random} \)
3: for \( i = 1 \) to \( T \) do
4: \( x, \hat{y} \sim D_{\text{generic}} \)
5: \( q, \hat{a} \leftarrow M^{-1}(\hat{y}) \)
6: \( \mathcal{L} \leftarrow -\log F_\phi(y|x, q, a) \)
7: \( \phi \leftarrow \text{MINIMIZE}(\mathcal{L}, \phi) \)
8: return \( F_\phi \)
9:
10: function GENERATE\((D_{\text{target}}, M, F_\phi)\)
11: \( D_{\text{synthetic}} \leftarrow [\ ] \)
12: for \((x, q, a) \in D_{\text{target}}\) do
13: \( \hat{y} \leftarrow \text{argmax} F_\phi(y|x, q, a) \)
14: if \( M(q, \hat{y}) = a \) then
15: append \((D_{\text{synthetic}}; (x, \hat{y})) \)
16: return \( D_{\text{synthetic}} \)

Splits: For the COCO and VQA datasets we use the ‘Karpathy’ splits from Karpathy and Li (2015). For GQA, we use the ‘balanced’ splits, but limit to \(~5K\) images each for the new test and dev sets from the original GQA dev set. Both the Visual7W and GQA datasets have images from Visual Genome (Krishna et al., 2016), and thus some (partially) overlap with the ‘Karpathy’ COCO images. Since we use COCO for pre-training \((D_{\text{generic}})\), we avoid data leakage by mapping Visual Genome IDs to COCO IDs and either filter questions about images that are in the COCO train or dev sets, or re-
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train Images</th>
<th>Train QA</th>
<th>Development Images</th>
<th>Development QA</th>
<th>Test Images</th>
<th>Test QA</th>
</tr>
</thead>
<tbody>
<tr>
<td>I  COCO</td>
<td>113,287</td>
<td>—</td>
<td>5,000</td>
<td>—</td>
<td>5,000</td>
<td>—</td>
</tr>
<tr>
<td></td>
<td>CapVQA</td>
<td>104,311</td>
<td>297,484</td>
<td>4,617</td>
<td>13,081</td>
<td>4,615</td>
</tr>
<tr>
<td></td>
<td>CapGQA</td>
<td>69,450</td>
<td>611,102</td>
<td>5,000</td>
<td>43,015</td>
<td>4,739</td>
</tr>
<tr>
<td></td>
<td>CapVisual7W</td>
<td>20,268</td>
<td>93,878</td>
<td>3,448</td>
<td>16,314</td>
<td>4,892</td>
</tr>
<tr>
<td></td>
<td>CapVizWiz</td>
<td>10,027</td>
<td>10,027</td>
<td>960</td>
<td>960</td>
<td>1,905</td>
</tr>
</tbody>
</table>

Table D.1: Statistics of the datasets used in this paper. Type I: generic/no QA. Type II: target/QA.

assign the data to match COCO. Finally, on VizWiz, we combine all of the original data and randomly re-partition it into test and dev sets of ~1K and ~2K images each, keeping the rest for training.