Towards VQA Models That Can Read

Amanpreet Singh[1], Vivek Natarajan, Meet Shah[1], Yu Jiang[1], Xinlei Chen[1],

Dhruv Batra[1,2], Devi Parikh[1,2], and Marcus Rohrbach[1]

[1]Facebook AI Research, [2]Georgia Institute of Technology

Abstract

Studies have shown that a dominant class of questions asked by visually impaired users on images of their surroundings involves reading text in the image. But today's VQA models can not read! Our paper takes a first step towards addressing this problem. First, we introduce a new "TextVQA" *dataset to facilitate progress on this important problem. Existing datasets either have a small proportion of questions about text (e.g., the VQA dataset) or are too small (e.g., the VizWiz dataset). TextVQA contains 45,336 questions on 28,408 images that require reasoning about text to answer. Second, we introduce a novel model architecture that reads text in the image, reasons about it in the context of the image and the question, and predicts an answer which might be a deduction based on the text and the image or composed of the strings found in the image. Consequently, we call our approach* Look, Read, Reason & Answer *(LoRRA). We show that LoRRA outperforms existing state-of-the-art VQA models on our TextVQA dataset. We find that the gap between human performance and machine performance is significantly larger on TextVQA than on VQA 2.0, suggesting that TextVQA is well-suited to benchmark progress along directions complementary to VQA 2.0.*

1. Introduction

The focus of this paper is endowing Visual Question Answering (VQA) models a new capability – the ability to *read text in images and answer questions* by reasoning over the text and other visual content.

VQA has witnessed tremendous progress. But today's VQA models fail catastrophically on questions requiring reading![1] This is ironic because these are *exactly* the questions visually-impaired users frequently ask of their assistive devices. Specifically, the VizWiz study [5] found that up to 21% of these questions. involve reading and reasoning about the text captured in the images of a user's surroundings – *'what temperature is my oven set to?'*, *'what denomination is this bill?'*.

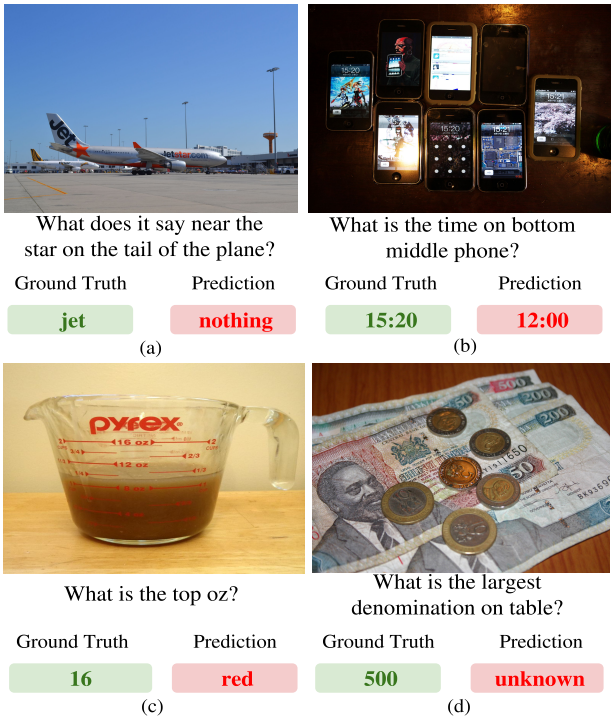


Figure 1: Examples from our TextVQA dataset that require VQA models to understand text embedded in images to answer the questions correctly. In green are ground truth answers and in red are answers predicted by a state-of-the-art VQA model (Pythia [17]). Clearly, today's VQA models fail at answering questions that involve reading and reasoning about text in images.

Consider the question in Fig. 1(a) – *'What does it say*

[1]All top entries in the CVPR VQA Challenges (2016-18) struggle to answer questions in category requiring reading correctly.

near the star on the tail of the plane?' from the TextVQA dataset. With a few notable exceptions, today's state-of-art VQA models are predominantly monolithic deep neural networks (without any specialized components). Consider what we are asking such models to learn to answer this question – the model must learn to

- realize when the question is about text (*'What . . . say?'*),
- detect image regions containing text (*'15:20'*, *'500'*),
- convert pixel representations of these regions (convolutional features) to symbols (*'15:20'*) or textual representations (semantic word-embeddings),
- jointly reason about detected text and visual content, *e.g*. resolving spatial or other visual reference relations (*'tail of the plane . . . on the back'*) to focus on the correct regions,
- finally, decide if the detected text needs to be 'copy-pasted' as the answer (*e.g*. *'16'* in Fig. 1 (c)) or if the detected text informs the model about an answer in the answer space (*e.g*. answering *'jet'*, in Fig. 1(a))

When laid out like that, it is perhaps unsurprising why today's models have not been able to make progress on reading questions – simply put, despite all the strengths of deep learning, it seems hopelessly implausible that all of the above skills will simply *emerge* in a monolithic network all from the distant supervision of VQA accuracy.

Fortunately, we can do more than just hope. Optical Character Recognition (OCR) is a mature sub-field of computer vision. The key thesis of this work is the following – we should bake in inductive biases and specialized components (*e.g*. OCR) into models to endow them with the different skills (*e.g*. reading, reasoning) required by the all-encompassing task of VQA.

Specifically, we propose a new VQA model that includes OCR as a module. We call it *Look, Read, Reason & Answer* (LoRRA). Our model architecture incorporates the regions (bounding boxes) in the image containing text as entities to attend over (in addition to object proposals). It also incorporates the actual text recognized in these regions (*e.g*. *'15:20'*) as information (in addition to visual features) that the model learns to reason over. Finally, our model includes a mechanism to decide if the answer produced should be 'copied' over from the OCR output (in more of a generation or slot-filling flavor), or should be deduced from the text (as in a standard discriminative prediction paradigm popular among existing VQA models). Our model learns this mechanism end-to-end. While currently limited in scope to OCR, our model is as an initial step towards endowing VQA models with the ability to reason over unstructured sources of external knowledge (in this case text found in a test image) and accommodate multiple streams of information flow (in this case predicting an answer from a predetermined vocabulary or generating an answer via copy).

One reason why there has been limited progress on VQA models that can read and reason about text in images is because such questions, while being a dominant category in real applications for aiding visually impaired users [5], are infrequent in the standard VQA datasets [3, 10, 51] because they were not collected in settings that mimic those of visually impaired users. While the VizWiz dataset [13] does contain data collected from visually impaired users, the effective size of the dataset is currently small due to 58% of the questions being "unanswerable" or images being "unsuitable" to answer. This makes it challenging to study the problem systematically, train effective models, or even draw sufficient attention to this important skill that current VQA models lack.

To this end, we introduce the TextVQA dataset. It contains 45,336 questions asked by (sighted) humans on 28,408 images from the Open Images dataset [27] from categories that tend to contain text *e.g*. "billboard", "traffic sign", "whiteboard". Questions in the dataset require reading and reasoning about text in the image. Each question-image pair has 10 ground truth answers provided by humans.

Models that do well on this dataset will not only need to parse the image and the question as in traditional VQA, but also read the text in the image, identify which of the text might be relevant to the question, and recognize whether a subset of the detected text is directly the answer (e.g., in the case of *'what temperature is my oven set to?'*) or additional reasoning is required on the detected text to answer the question (e.g., *'which team is winning?'*).

We show that LoRRA outperforms existing state-of-the-art VQA models on the TextVQA dataset. Overall, our contributions are:

- We introduce a novel dataset (TextVQA) containing questions which require the model to read and reason about the text in the image to be answered.
- We propose *Look, Read, Reason & Answer* (LoRRA): a novel model architecture which explicitly reasons over the outputs from an OCR system when answering the questions.
- LoRRA outperforms existing state-of-the-art VQA models on our TextVQA dataset.

2. Related work

Visual Question Answering. VQA has seen numerous advances and new datasets since the first large-scale VQA dataset was introduced by Antol *et al*. [3]. This dataset was larger, more natural, and more varied than earlier VQA datasets such as DAQUAR [31] or COCO-QA [38] but had linguistic priors which were exploited by models to answer questions without sufficient visual grounding. This issue was addressed by Goyal *et al*. [10] by adding complementary triplets (I_c, q, a_c) for each original triplet (I_o, q, a_o) where image I_c is similar to image I_o but the answer for the given question q changes from a_o to a_c. To study visual reasoning independent of language, non-photo-realistic

VQA datasets have been introduced such as CLEVR [18], NLVR [42] and FigureQA [21]. Wang *et al*. [45] introduced a Fact-Based VQA dataset which explicitly requires external knowledge to answer a question.

Text based VQA. Several existing datasets study text detection and/or parsing in natural everyday scenes: COCO-Text [43], Street-View text [44] IIIT-5k [33] and ICDAR 2015 [22]. These do not involve answering questions about the images or reasoning about the text. DVQA [20] assesses automatic bar-chart understanding by training models to answer questions about graphs and plots. The Multi-Output Model (MOM) introduced in DVQA uses an OCR module to read chart specific content. Textbook QA (TQA) [24] considers the task of answering questions from middle-school textbooks, which often require understanding and reasoning about text and diagrams. Similarly, AI2D [23] contains diagram based multiple-choice questions. Note that these works all require reasoning about text to answer questions, but in narrow domains (bar charts, textbook diagrams, etc.). The focus of our work is to reason and answer questions about text in natural everyday scenes. MemexQA [16] introduces VQA on a collection of photos and videos which often involves reasoning about the time and date at which the photo was taken, but this information is structured and is provided as part of the meta data.

Visual Representations for VQA Models. VQA models typically use some variant of attention to get a representation of the image that is relevant for answering the question [2, 7, 30, 47, 48, 51, 17]. Object region proposals and the associated features are generated by using a detection network which are then spatially attended to, conditioned on a question representation. In this work, we extend the representations that a VQA model reasons over. Specifically, in addition to attending over object proposals, our model also attends over the regions where text is detected.

Copy Mechanism. A core component of our proposed model is its ability to decide whether the answer to a question should be an OCR token detected in the image, or if the OCR tokens should only inform the answer to the question. The former is implemented as a "copy mechanism" – a learned slot filling approach. Our copy mechanism is based on a series of works in NLP on pointer generator networks [11, 39, 32, 12, 34]. A copy mechanism provides networks the ability to generate out-of-vocabulary words by pointing at a word in context and then copying it to the generated result. Such approaches have been used for a variety of tasks in NLP such as summarization [11, 34, 39], question answering [46], language modelling [32], neural machine translation [12], and dialog [37].

3. LoRRA: Look, Read, Reason & Answer

In this section we introduce our novel model architecture to answer questions about images which require reading the text in the images to answer the questions.

We assume we get an image v and a question q as input, where the question consists of L words $w_1, w_2, \ldots, w_L$. At a high level, our model contains three components: (i) a **VQA component** to reason and infer the answer based on the image v and the question q (Sec 3.3); (ii) a **reading component** which allows our model to read the text present in the image (Sect 3.2); and (iii) an **answering module** which either predicts from an answer space or points to the text read by the *reading component* (Sec. 3.3). The overall model is shown in Fig. 2. Note that, the OCR module and backbone VQA model and can be any OCR model and any recent attention-based VQA model. Our approach is agnostic to the internal details of these components. Finally, we detail the exact implementation choices and hyper parameters in Sec. 3.4.

3.1. VQA Component

Similar to many state-of-the-art VQA models [7, 17], we first embed the question words $w_1, w_2, \ldots, w_L$ of the question q with a pre-trained embedding function (*e.g*. GloVe [36]) and then encode the resultant word embeddings iteratively with a recurrent network (*e.g*. LSTM [15]) to produce a question embedding $f_Q(q)$. For images, the visual features are represented as spatial features, either in the form of grid-based convolutions and/or features extracted from the bounding box proposals [1]. We refer to these features as $f_I(v)$ where f_I is the network which extracts the image representation. We use an attention mechanism f_A over the spatial features [4, 7], which predicts attentions based on the $f_I(v)$ and $f_Q(q)$ and gives a weighted average over the spatial features as the output. We then combine the output with the question embedding. At a high level, the calculation of our VQA features $f_{VQA}(v, q)$ can be written as:

$$f_{VQA}(v, q) = f_{comb}(f_A(f_I(v), f_Q(q)), f_Q(q)) \quad (1)$$

where f_{comb} is the combination module, visualized as the left $\bigotimes$ in Fig. 2.

Assuming we have a fixed answer space of $a_1, \ldots, a_N$, we use a feed-forward MLP f_c on the combined embedding $f_{VQA}(v, q)$ to predict probabilities $p_1, \ldots, p_N$ where the probability of a_i being the correct answer is p_i.

3.2. Reading Component

To add the capability of reading text from an image, we rely on an OCR model which is not jointly trained with our system. We assume that the OCR model can read and return word tokens from an image, e.g. [6, 41]. The OCR model extracts words $s = s_1, s_2, ..., s_M$ from the images, where M is the number of tokens. We then embed the OCR tokens with a pre-trained word embedding, f_O. Finally, we use the same architecture as the VQA component to get combined

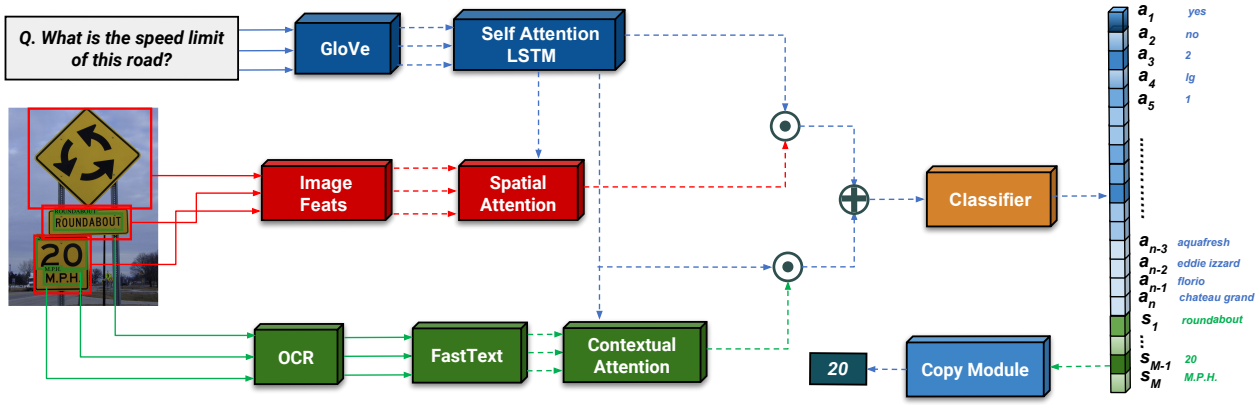


Figure 2: Overview of our approach *Look, Read, Reason & Answer* (LoRRA). Our approach looks at the image, reads its text, reasons about the image and text content and then answers, either with an answer a from the fixed answer vocabulary or by selecting one of the OCR strings s. Dashed lines indicated not jointly-trained components. The blocks with darker color have more attention weight. "20" OCR token has highest attention weight in the example.

VQA 2.0 Accuracy		
Model	**test-dev**	**test-std**
BUTD [1]	65.32	65.67
Counter [50]	68.09	68.41
BAN [25]	69.08	69.50
BAN [25] + Counter [50]	70.04	70.35
Pythia v0.1 [17]	70.01	70.24
Pythia v0.3 (Ours)	70.21	70.33
VizWiz Accuracy		
Model	**test**	
BAN[25]	51.40	
Ours	54.72	

Table 1: **Single model VQA 2.0 and VizWiz performance in %**. The VQA component of our model outperforms or is comparable to prior state-of-the-art.

OCR-question features, f_{OCR}. Specifically,

$$f_{OCR}(s,q) = f_{comb}(f_A(f_O(s), f_Q(q)), f_Q(q)) \quad (2)$$

This is visualized in Fig. 2 (right). Note that the parameters of the functions f_A and f_{comb} are not shared with the VQA model component above but they have the same architecture, just with different input dimensions. During weighted attention as the features are multiplied by weights and then averaged, the ordering information gets lost. To provide the answer module with the ordering information of the original OCR tokens, we concatenate the attention weights and the final weight-averaged features. This allows the answer module to know the original attention weights for each token in order.

3.3. Answer Module

With a fixed answer space, the current VQA models are only able to predict fixed tokens which limits the generalization to out-of-vocabulary (OOV) words. As the text in images frequently contains words not seen at training time, it is hard to answer text-based questions based on a pre-defined answer space alone. To generalize to arbitrary text, we take inspiration from pointer networks which allow pointing to OOV words in context [11, 39, 32, 12, 34]. We extend our answer space through addition of a dynamic component which corresponds to M OCR tokens. The model now has to predict probabilities $(p_1, \ldots, p_N, \ldots, p_{N+M})$ for $N+M$ items in the answer space instead of the original N items.

We pick the index with the highest probability p_i as the index of our predicted answer. If the model predicts an index larger than N (i.e., among the last M tokens in answer space), we directly *"copy"* the corresponding OCR token as the predicted answer. Hence, our answering module can be thought of as *"copy if you need"* module which allows answering from the OOV words using the OCR tokens.

Collecting all the components, the final equation f_{LoRRA} for predicting the answer probabilities can be written as:

$$f_{LoRRA}(v,s,q) = f_{MLP}([f_{VQA}(v,q); f_{OCR}(s,q)]) \quad (3)$$

where $[;]$ refers to concatenation and f_{MLP} is a two-layer feed-forward network which predicts the binary probabilities as logits for each answer. We opt for binary cross entropy using logits instead of calculating the probabilities through softmax as it allows us to handle cases where the answer can be in both the actual answer space and the OCR tokens without penalizing for predicting either one

(a) Question: which processor Brand is featured on the top left?
Answer: intel

(b) Question: which brand are the crayons?
Answer: crayola

(c) Question: what is the name of the bose speaker style in these boxes?
Answer: freestyle

(d) Question: what is the license number?
Answer: cu58 ckk

Figure 3: Examples from TextVQA. Questions require inferring hidden characters ("intel"), handling rotated text ("crayola"), reasoning ("bose" vs "freestyle") and selecting among multiple texts in image "cu58 ckk" vs "western power distribution").

(the likelihood of logits is independent of each other). Note that if the model chooses to copy, it can only produce one of the OCR tokens as the predicted answer. 8.9% of the TextVQA questions can be answered only by combining multiple OCR tokens; we leave this as future work.

3.4. Implementation Details

We base our VQA component on the VQA 2018 challenge winner entry, Pythia [17]. Our re-implementation with slight changes in hyper-parameters (*e.g.* size of question vocabulary, hidden dimensions) achieves state-of-the-art VQA accuracy for a single model (*i.e.* w/o ensemble) as shown in Table 1 on both the VQA v2.0 dataset [9], as well as the VizWiz dataset [13].

Pythia [17][40] was inspired from the detector-based bounding box prediction of the bottom-up top-down attention network [1] (VQA winner 2017), which in turn has a multi-modal attention mechanism similar to the VQA 2016 winner [7], which relied on grid-based features. Following Pythia, for spatial features $f_I(v)$, we rely on both grid and region based features for the input image. The grid based features are obtained by average pooling $2048\mathcal{D}$ features from the `res-5c` block of a pre-trained ResNet-152 [14]. The region based features are extracted from `fc6` layer of an improved Faster-RCNN model [8] trained on the Visual Genome [28] objects and attributes as provided in [1]. During training, we fine-tune the `fc7` weights as in [17].

We use pre-trained GloVe embeddings with a custom vocabulary (top ∼77k question words in the VQA 2.0 dataset) for the question embedding [36]. The f_Q module passes GloVe embeddings to an LSTM [15] with self-attention [49] to generate a single question embedding for the question. For OCR, we run the Rosetta OCR system [6] to provide us word strings $s_1, ..., s_N$. OCR tokens are first embedded using pretrained FastText embeddings (f_O) [19], which can generate word embeddings even for OOV tokens.

In f_A, the question embedding $f_Q(q)$ is used to obtain the top-down *i.e.* task-specific attention on both $f_O(s)$ OCR tokens features and $f_I(v)$ image features. The features are then averaged based on the attention weights to get a final feature representation for both the OCR tokens and the image features. The final grid-level and region-based features are concatenated in case of image features. For the OCR tokens, attention weights are concatenated to the final attended features as explained in 3.1. Finally, in $f_{comb}(x, y)$, the two feature embeddings in consideration are fused using element-wise/hadamard product, $\otimes$, of the features. The fused features from $f_{OCR}(s, q)$ and $f_{VQA}(v, q)$ are concatenated and passed through an MLP to produce logits.

4. TextVQA

To study the task of answering questions that require reading text in images, we collect a new dataset called TextVQA which is publicly available at https://textvqa.org. In this section, we start by describing how we selected the images that we use in TextVQA. We then explain our data collection pipeline for collecting the questions and the answers. Finally, we provide statistics and an analysis of the dataset. Snapshots of the annotation interface and detailed instructions can be found in the supplementary material.

4.1. Images

We use Open Images v3 dataset [27] as the source of our images. In line with the goal of developing and studying VQA models that reason about text, we are most interested in the images that contain text in them. Several categories in Open Images fit this criterion (*e.g.*, billboard, traffic sign, whiteboard). To automate this process of identifying categories that tend to have images with text in them, we select 100 random images for each category (or all images if max images for that category is less than 100). We run a state-of-the-art OCR model Rosetta [6] on these images and compute the average number of OCR boxes in a category. The average number of OCR boxes per-category were normalized and used as per-category weights for sampling the

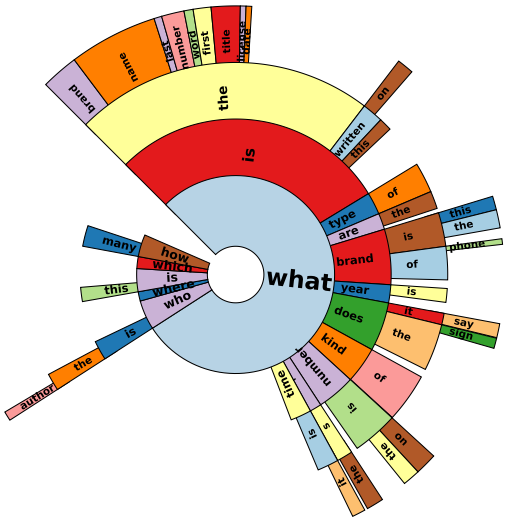


Figure 4: Distribution of first four words in questions in TextVQA. Most questions start with "what".

images from the categories.

Figure 5: (Left) Wordcloud for majority answers in TextVQA. Frequently occurring answers include yes, brand names, "stop" and city names. (Right) Wordcloud for OCR tokens predicted by the OCR model we use. Note the overlap with answers on brand names *(lg)*, cities *(london)* and verbs *(stop)*.

We collect TextVQA's training and validation set from Open Images' training set while test set is collected from Open Images' test set. We set up a three stage pipeline for crowd-sourcing our data. In the first stage, annotators were asked to identify images that did not contain text (using a forced-choice "yes/"no" flag). Filtering those (and noisy data from annotators) out resulted in 28,408 images, which from the basis of our TextVQA dataset.

4.2. Questions and Answers

In the second stage, we collect 1-2 questions for each image. To collect the first question, we show annotators an image and ask them to provide a question which requires reading the text to answer. Specifically, they were told to *'Please ensure that answering the question requires reading of the text in the image. It is OK if the answer cannot be directly copied from the text but needs to be inferred or paraphrased.'*

To collect a second question that is different from the first, we show annotators the first question and ask them to come up with a question that requires reasoning about the text in the image and has a different answer.

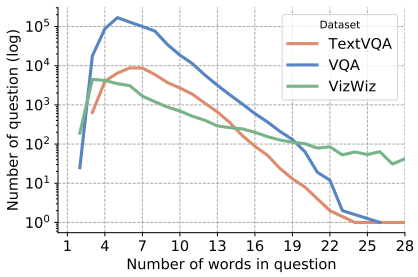
Following VQA [3, 10] and VizWiz [13] datasets, we collect multiple answers for each question. We have 10 answers for each question.

To ensure answer quality, we gave annotators instructions similar to those used in [3, 13] when collecting the VQA and VizWiz datasets. In addition, to catch any poor quality data from earlier steps, we give annotators these four options: (i) no text in image; (ii) not a question; (iii) answering the question doesn't require reading any text in image; and (iv) unanswerable, *e.g.* questions involving speculation about the meaning of text. We remove the questions where a majority of workers marked any of these flags. Additionally, we use hand-crafted questions for which we know the correct answers to identify and filter out bad annotators.

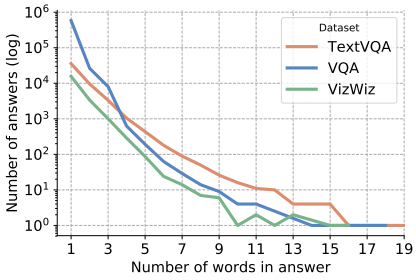
4.3. Statistics and Analysis

We first analyze the diversity of the questions that we have in the dataset. TextVQA contains 37,912 unique questions which is 83.6% of the entire dataset. Fig. 6a shows the distribution of question length. We also show the same stats for the VQA 2.0 and the VizWiz datasets for reference. The average question length in TextVQA is 7.18 words and slightly higher than in VQA 2.0 (6.29) and VizWiz (6.68). We also note that TextVQA does not contain questions with length less than 3 words. Workers often form questions which are longer to disambiguate the response (*e.g.* specifying where exactly the text is in the image, see Figure 3). Fig. 6d shows top 15 most occurring questions in the dataset with their count while Fig. 6e shows top 500 most occurring questions and their counts. We can see the uniform shift from common question about "time" to questions occurring in specific situations like "team names". Fig. 4 shows sunburst for first 4 words in questions. We also observe that most questions involve reasoning about common things (*e.g.* figuring out brand names, cities and temperature). Questions often start with "what", frequently inquiring about "time", "names", "brands" or "authors".

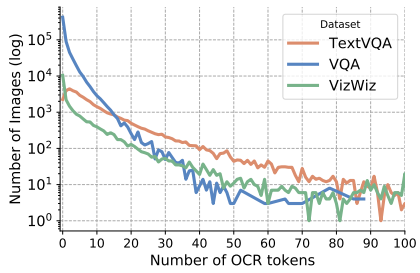
In total there are 26,263 (49.2%) unique majority answers in TextVQA. The percentage of unique answers in TextVQA is quite high compared to VQA 2.0 (3.4%) and VizWiz (22.8%). All 10 annotators agree on the most common answer for 22.8% questions, while 3 or more annotators agree on most common answer for 97.9% questions. Fig. 5 (left) shows a word cloud plot for the majority answers in the dataset. The answer space is diverse and involves brand names, cities, people's names, time, and countries. Note that this diversity makes it difficult to have a fixed answer space – a challenge that most existing VQA datasets do not typically pose. The most common answer



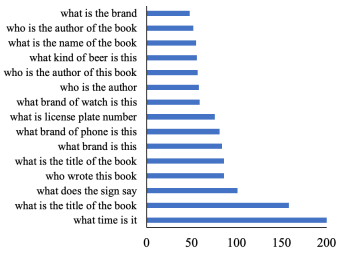
(a) **Number of questions with a particular question length**. We see that average question length (7.16) is higher in TextVQA compared to others.



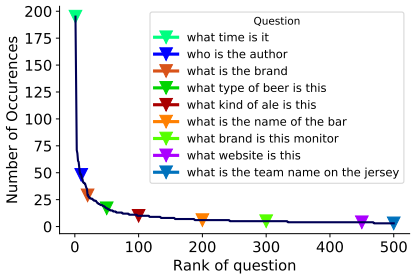
(b) **Number of majority answers with a particular length**. Average answer length (1.7) is high and answer can contain long paragraph and quotes.



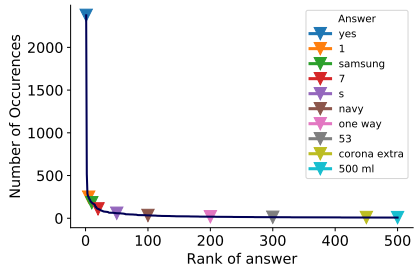
(c) **Number of images with a particular number of OCR tokens**. Average number of tokens is around 3.14. In TextVQA, 10x more images contain OCR text than others.



(d) **Text of top 15 most occurring questions in TextVQA.** Most of the top questions start from "what".



(e) **Total occurrences for 500 most common questions** among 23184 unique questions along with some pointer markers to questions at particular ranks.



(f) Similar to 6e, plot shows **total occurrences for 500 most common majority answers** along with pointer markers to answers at particular ranks.

Figure 6: **Question, Answer and OCR statistics for TextVQA**. We show comparisons with VQA 2.0 [10] and VizWiz [13].

("yes") is the majority answer for only 4.71% of the dataset and "yes/no" (majority answer) questions in total only make up 5.55% of the dataset. The average answer length is 1.58 (Fig. 6b). In a few occurrences where the text in the image is long (e.g., a quote or a paragraph), the answer length is high. Fig. 6f shows the frequency of top 500 most common answers. The gradual shift from brands to rare cities is depicted. We also note that the drop in TextVQA for number of answers of a particular answer length is more gradual than in VQA 2.0 which drops sharply after answer length 3.

Finally, we analyze the OCR tokens produced by the Rosetta OCR system [6]. In Fig. 6c, we plot number of images containing "x" number of OCR tokens. The peak between 4 and 5 shows that a lot of images in our dataset contain a good number of OCR tokens. In some cases, when the system is unable to detect text we get 0 tokens but those cases are restricted to ∼1.5k images and we manually verified that the images actually do contain text. Fig. 5 (right) shows a word cloud of OCR tokens. We see that OCR tokens do contain common answers such as brand names and cities.

5. Experiments

We conduct our experiments on TextVQA and provide both validation and test accuracies. We start by explaining our baselines including both heuristics and end-to-end trained models which we compare with LoRRA. We divide TextVQA in train, validation and test splits with size 34,602, 5,000, 5,734. The TextVQA questions collected from Open Images v3's training set were randomly split into training and validation set. There is no image overlap between the sets. For our approach, we use a vocabulary **SA** of size 3996, which contains answers which appear at least twice in the training set. For the baselines that don't use the copy mechanism, this vocabulary turns out to be too limited. To give them a fair shot, we also create a larger vocabulary (LA), containing the 8000 most frequent answers (including those that occur just once).

Upper Bounds and Heuristics. These mainly evaluate the upper bounds of what can be achieved using the OCR tokens detected by our OCR module, and benchmark any biases in the dataset. We test (1) **OCR UB:** the upper bound accuracy one can get if the answer can be build directly from OCR tokens (and can always be predicted correctly). **OCR UB** considers combinations of OCR tokens upto 4-

Model	Accuracy(%) Val	Test
Human	85.01	86.79
OCR UB	37.12	36.52
Rand 100	0.22	0.20
Wt. Rand 100	0.27	0.26
Majority Ans	4.48	2.63
Random OCR	7.72	9.12
OCR Max	9.76	11.60

Model	Vocab	Accuracy(%) Val	Test
Q	LA	8.09	8.70
I	LA	6.29	5.58
I+Q (Pythia)	LA	13.04	14.0
I+Q+O (ours)	LA	18.35	–
I+Q+O+C (ours)	n/a	20.06	–
LoRRA+Pythia (ours)	LA	26.23	–
LoRRA+Pythia (ours)	SA	**26.56**	**27.63**

Table 2: Evaluation on TextVQA. (Left) Accuracies for various heuristics baselines, which show that using **OCR** can help in achieving a good accuracy on TextVQA. (Right) Accuracies of our trained baselines and ablations in comparison with our model **LoRRA**. **I** denotes usage of image features, **Q** question features, **O** OCR tokens' features, and **C** copy mechanism. Our model **LoRRA** outperforms VQA SoTA (I+Q, Pythia) and other baselines.

grams. (ii) **Rand 100:** the accuracy one can get by selecting a random answer from the vocabulary of top 100 most frequent answers (iii) **Wt. Rand 100:** the accuracy of baseline (ii) but with weighted random sampling using the frequency of the 100 most occurring tokens as weights. (iv) **Majority Ans:** the accuracy of always predicting the majority answer "yes" (v) **Random OCR token:** the accuracy of predicting a random OCR token from the list of OCR tokens detected in an image (vi) **OCR Max:** accuracy of always predicting the OCR token that occurs most frequently in the image (e.g., in the picture of a book on Anatomy, the word "Anatomy" may occur multiple times, and there is a good chance the question is one whose answer is "Anatomy").

Baselines. All of the baselines use **LA** unless specified otherwise. We make modifications to the implementation discussed in Sec. 3.4 to generate our baselines which include (i) **Question Only (Q):** we only use the $f_Q(q)$ module of LoRRA to predict the answer. Specifically, the rest of the features are zeroed out. (ii) **Image Only (I):** similar to **Q**, we only use image features $f_I(v)$ to predict answers. **Q** and **I** do not have access to OCR tokens and predict against the default answer space.

Ablations. We create several ablations of our approach LoRRA by using the reading component and answering module in conjunction and alternatively. (i) **I+Q:** LoRRA with no OCR features. This ablation is state-of-the-art for VQA 2.0, see Pythia v0.3 in Tab. 1; (ii) **I+Q+O:** (i) with OCR features as input but no copy module or dynamic answer space; (iii) **I+Q+O+C:** (ii) but with the copy mechanism and no fixed answer space *i.e.* the model can only predict from the OCR tokens. We use the abbreviation **C** when we add the copy module and dynamic answer space to a model.

Our full model **LoRRA** corresponds to I+Q+O+C. We also compare **LoRRA** with small answer space (SA) to a version with large answer space (LA).

Experimental Setup. We develop our model in PyTorch [35]. We use AdaMax optimizer [26] to perform back-propagation [29]. We predict logits and train using binary cross-entropy loss for them. We train all of our models for 24000 iterations with a batch size of 128 on 8 GPUs. We set the maximum question length to 14 and maximum number of OCR tokens to 50. We pad the rest of the sequence if it is less than maximum length. We use the VQA accuracy metric for evaluation [10]. During training, we keep an exponential moving average with decay of 0.99. We use a learning rate of 5e-2 for all layers except the $fc7$ layers used for fine-tuning which are trained with 5e-3. We uniformly decrease the learning rate to 5e-4 after 14k iterations. We calculate val accuracy at every 1000th iteration and use the model with the best validation accuracy to calculate test accuracy.

Results. Tab. 2 shows accuracies on both heuristics (left) and trained baselines and models (right). Despite collecting open-ended answers from annotators, we find that human accuracy is 85.01%, consistent with that on VQA 2.0 [10] and VizWiz [13]. While the OCR system we used is not perfect, the upper-bound on the validation set that one can achieve by correctly predicting the answer using these OCR tokens is 37.12%. This is higher than our best model, suggesting room for improvement to reason about the OCR tokens. Majority answer ("yes") gets only 4.48% on test set. Random baselines, even the weighted one, are rarely correct. **Random OCR** token selection and maximum occurring OCR token selection (**OCR Max**) yields better accuracies compared to other heuristics baselines.

Question only **(Q)** and Image only **(I)** baseline get 8.09% and 6.29% validation accuracies respectively which shows that the dataset does not have significant biases w.r.t. images and questions. Our VQA model, **I+Q** (Pythia v0.3), which is an improvement over the original Pythia [40] model, is state-of-the-art on VQA 2.0 and VizWiz, but only achieves 13.04% validation accuracy on TextVQA. This demonstrates the inability of current VQA models to read and reason about text in images. A jump in accuracy to 18.35% is observed by feeding OCR tokens **(I+Q+O)** into the model; this supports the hypothesis that OCR tokens do help in predicting correct answers. Validation accuracy of 20.06 achieved by **I+Q+O+C** by only using OCR tokens to predict answers without using any fixed answer space, further bolsters OCR importance as it is quite high compared to our Pythia[40] 0.3. Our LoRRA (LA) model outperforms all of the ablations. Finally, a slight modification which allows the model to predict from the OCR tokens more often by changing the fixed answer space **LA** to **SA** further improves performance.

While LoRRA can reach up to 26.56% accuracy on the TextVQA's validation set, there is a large gap to human per-

formance of 85.01%.

6. Conclusion

We explore a specific skill in Visual Question Answering that is critical for applications involving aiding visually impaired users – answering questions about everyday images that involve reading and reasoning about text in these images. We find that existing datasets do not support a systematic exploration of the research efforts towards this goal. To this end, we introduce the TextVQA dataset which contains questions which can only be answered by reading and reasoning about text in images. We also introduce *Look, Read, Reason & Answer* (LoRRA), a novel model architecture for answering questions based on text in images. LoRRA reads the text in images, reasons about it based on the provided question, and predicts an answer from a fixed vocabulary or the text found in the image. LoRRA is agnostic to the specifics of the underlying OCR and VQA modules. LoRRA significantly outperforms the current state-of-the-art VQA models on TextVQA. Our OCR model, while mature, still fails at detecting text that is rotated , a bit unstructured (e.g., a scribble) or partially occluded. We believe TextVQA will encourage research both on improving text detection and recognition in unconstrained environments, as well as in enabling the VQA models to read and reason about text in images.

References

[1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[2] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *CVPR*, 2016.

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[5] J. P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. C. Miller, R. Miller, A. Tatarowicz, B. White, S. White, et al. Vizwiz: nearly real-time answers to visual questions. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 333–342. ACM, 2010.

[6] F. Borisyuk, A. Gordo, and V. Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 71–79. ACM, 2018.

[7] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *arXiv preprint arXiv:1606.01847*, 2016.

[8] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. Detectron. `https://github.com/facebookresearch/detectron`, 2018.

[9] P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He. Accurate, large minibatch sgd: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[10] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] J. Gu, Z. Lu, H. Li, and V. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Annual Meeting of the Association for Computational Linguistics (ACL), 2016*. Association for Computational Linguistics., 2016.

[12] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio. Pointing the unknown words. *arXiv preprint arXiv:1603.08148*, 2016.

[13] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. *arXiv preprint arXiv:1802.08218*, 2018.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] L. Jiang, J. Liang, L. Cao, Y. Kalantidis, S. Farfade, and A. G. Hauptmann. Memexqa: Visual memex question answering. *arXiv:1708.01336*, 2017.

[17] Y. Jiang, V. Natarajan, X. Chen, M. Rohrbach, D. Batra, and D. Parikh. Pythia v0. 1: the winning entry to the vqa challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018.

[18] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 1988–1997. IEEE, 2017.

[19] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[20] K. Kafle, S. Cohen, B. Price, and C. Kanan. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5648–5656, 2018.

[21] S. E. Kahou, V. Michalski, A. Atkinson, A. Kadar, A. Trischler, and Y. Bengio. Figureqa: An annotated figure dataset for visual reasoning. *arXiv preprint arXiv:1710.07300*, 2017.

[22] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, et al. Icdar 2015 competition on robust reading. In *Document Analysis and Recognition (ICDAR), 2015 13th International Conference on*, pages 1156–1160. IEEE, 2015.

[23] A. Kembhavi, M. Salvato, E. Kolve, M. Seo, H. Hajishirzi, and A. Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision*, pages 235–251.

Springer, 2016.

[24] A. Kembhavi, M. J. Seo, D. Schwenk, J. Choi, A. Farhadi, and H. Hajishirzi. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *CVPR*, volume 2, page 3, 2017.

[25] J.-H. Kim, J. Jun, and B.-T. Zhang. Bilinear attention networks. *arXiv preprint arXiv:1805.07932*, 2018.

[26] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[27] I. Krasin, T. Duerig, N. Alldrin, A. Veit, S. Abu-El-Haija, S. Belongie, D. Cai, Z. Feng, V. Ferrari, V. Gomes, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github. com/openimages*, 2(6):7, 2016.

[28] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017.

[29] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[30] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016.

[31] M. Malinowski and M. Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *Advances in neural information processing systems*, pages 1682–1690, 2014.

[32] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[33] A. Mishra, K. Alahari, and C. Jawahar. Scene text recognition using higher order language priors. In *BMVC-British Machine Vision Conference*. BMVA, 2012.

[34] R. Nallapati, B. Zhou, C. Gulcehre, B. Xiang, et al. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*, 2016.

[35] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[36] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

[37] D. Raghu, N. Gupta, et al. Hierarchical pointer memory network for task oriented dialogue. *arXiv preprint arXiv:1805.01216*, 2018.

[38] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *Advances in neural information processing systems*, pages 2953–2961, 2015.

[39] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017.

[40] A. Singh, V. Natarajan, Y. Jiang, X. Chen, M. Shah, M. Rohrbach, D. Batra, and D. Parikh. Pythia-a platform for vision & language research. *SysML Workshop, NeurIPS 2019*, 2018.

[41] R. Smith. An overview of the tesseract ocr engine. In *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, volume 2, pages 629–633. IEEE, 2007.

[42] A. Suhr, M. Lewis, J. Yeh, and Y. Artzi. A corpus of natural language for visual reasoning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 217–223, 2017.

[43] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016.

[44] K. Wang and S. Belongie. Word spotting in the wild. In *European Conference on Computer Vision*, pages 591–604. Springer, 2010.

[45] P. Wang, Q. Wu, C. Shen, A. Dick, and A. van den Hengel. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[46] C. Xiong, V. Zhong, and R. Socher. Dynamic coattention networks for question answering. *arXiv preprint arXiv:1611.01604*, 2016.

[47] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision*, pages 451–466. Springer, 2016.

[48] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *CVPR*, 2016.

[49] Z. Yu, J. Yu, C. Xiang, J. Fan, and D. Tao. Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[50] Y. Zhang, J. Hare, and A. Prügel-Bennett. Learning to count objects in natural images for visual question answering. *arXiv preprint arXiv:1802.05766*, 2018.

[51] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei. Visual7w: Grounded question answering in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4995–5004, 2016.