

Iterative Answer Prediction with Pointer-Augmented Multimodal Transformers for TextVQA

Ronghang Hu^{1,2} Amanpreet Singh¹

¹Facebook AI Research (FAIR)

{ronghang, trevor}@eecs.berkeley.edu, {asg, mrf}@fb.com

Trevor Darrell² Marcus Rohrbach¹

²University of California, Berkeley

Abstract

Many visual scenes contain text that carries crucial information, and it is thus essential to understand text in images for downstream reasoning tasks. For example, a deep water label on a warning sign warns people about the danger in the scene. Recent work has explored the TextVQA task that requires reading and understanding text in images to answer a question. However, existing approaches for TextVQA are mostly based on custom pairwise fusion mechanisms between a pair of two modalities and are restricted to a single prediction step by casting TextVQA as a classification task. In this work, we propose a novel model for the TextVQA task based on a multimodal transformer architecture accompanied by a rich representation for text in images. Our model naturally fuses different modalities homogeneously by embedding them into a common semantic space where self-attention is applied to model inter- and intra-modality context. Furthermore, it enables iterative answer decoding with a dynamic pointer network, allowing the model to form an answer through multi-step prediction instead of one-step classification. Our model outperforms existing approaches on three benchmark datasets for the TextVQA task by a large margin.

1. Introduction

As a prominent task for visual reasoning, the Visual Question Answering (VQA) task [4] has received wide attention in terms of both datasets (*e.g.* [4, 17, 22, 21, 20]) and methods (*e.g.* [14, 3, 6, 25, 33]). However, these datasets and methods mostly focus on the visual components in the scene. On the other hand, they tend to ignore a crucial modality – text in the images – that carries essential information for scene understanding and reasoning. For example, in Figure 1, *deep water* on the sign warns people about the danger in the scene. To address this drawback, new VQA datasets [44, 8, 37] have been recently proposed with questions that explicitly require understanding and reasoning about text in the image, which is referred to as the TextVQA task.

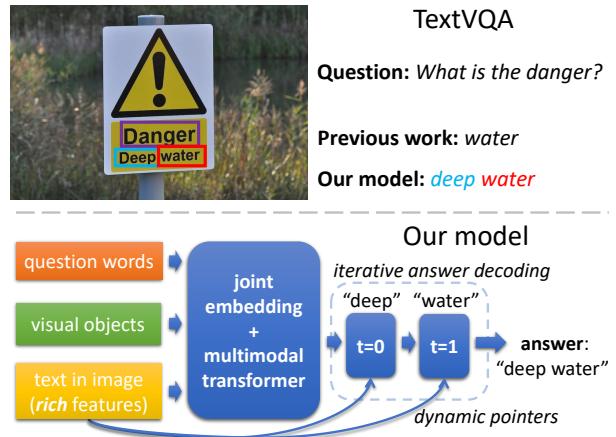


Figure 1. Compared to previous work (*e.g.* [44]) on the TextVQA task, our model, accompanied by rich features for image text, handles all modalities with a multimodal transformer over a joint embedding space instead of pairwise fusion mechanisms between modalities. Furthermore, answers are predicted through iterative decoding with pointers instead of one-step classification over a fixed vocabulary or copying single text token from the image.

The TextVQA task distinctively requires models to see, read and reason over three modalities: the input question, the visual contents in the image such as visual objects, and the text in the image. Several approaches [44, 8, 37, 7] have been proposed for the TextVQA task, based on OCR results of the image. In particular, LoRRA [44] extends previous VQA models [43] with an OCR attention branch and adds OCR tokens as a dynamic vocabulary to the answer classifier, allowing copying a single OCR token from the image as the answer. Similarly in [37], OCR tokens are grouped into blocks and added to the output space of a VQA model.

While these approaches enable reading text in images to some extent, they typically rely on custom pairwise multimodal fusion mechanisms between two modalities (such as single-hop attention over image regions and text tokens, conditioned on the input question), which limit the types of possible interactions between modalities. Furthermore, they treat answer prediction as a single-step classification problem – either selecting an answer from the training set

answers or copying a text token from the image – making it difficult to generate complex answers such as book titles or signboard names with multiple words, or answers with both common words and specific image text tokens, such as *McDonald’s burger* where *McDonald’s* is from text in the image and *burger* is from the model’s own vocabulary. In addition, the word embedding based image text features in previous work have limited representation power and miss important cues such as the appearance (*e.g.* font and color) and the location of text tokens in images. For example, tokens that have different fonts and are spatially apart from each other usually do not belong to the same street sign.

In this paper, we address the above limitations with our novel Multimodal Multi-Copy Mesh (M4C) model for the TextVQA task, based on the transformer [48] architecture accompanied by iterative answer decoding through dynamic pointers, as shown in Figure 1. Our model naturally fuses the three input modalities and captures intra- and inter-modality interactions homogeneously within a multimodal transformer, which projects all entities from each modality into a common semantic embedding space, and applies the self-attention mechanism [38, 48] to collect relational representations for each entity. Instead of casting answer prediction as a classification task, we perform iterative answer decoding in multiple steps and augment our answer decoder with a dynamic pointer network that allows selecting text in the image in a permutation-invariant way without relying on any ad-hoc position indices in previous work such as LoRRA [44]. Furthermore, our model is capable of combining its own vocabulary with text in the image in a generated answer, as shown in examples in Figure 4 and 5. Finally, we introduce a rich representation for text tokens in the images based on multiple cues, including its word embedding, appearance, location, and character-level information.

Our contributions in this paper are as follows: 1) We show that multiple (more than two) input modalities can be naturally fused and jointly modeled through our multimodal transformer architecture. 2) Unlike previous work on TextVQA, our model reasons about the answer beyond a single classification step and predicts it through our pointer-augmented multi-step decoder. 3) We adopt a rich feature representation for text tokens in images and show that it is better than features based only on word embedding in previous work. 4) Our model significantly outperforms previous work on three challenging datasets for the TextVQA task: TextVQA [44] (+25% relative), ST-VQA [8] (+65% relative), and OCR-VQA [37] (+32% relative).

2. Related work

VQA based on reading and understanding image text. Recently, a few datasets and methods [44, 8, 37, 7] have been proposed for visual question answering based on text in images (referred to as the TextVQA task). LoRRA [44],

a prominent prior work on this task, extends the Pythia [43] framework for VQA and allows it to copy a single OCR token from the image as the answer, by applying a single attention hop (conditioned on the question) over the OCR tokens and including the OCR token indices in the answer classifier’s output space. A conceptually similar model is proposed in [37], where OCR tokens are grouped into blocks and added to both the input features and the output answer space of a VQA model. In addition, a few other approaches [8, 7] enable text reading by augmenting existing VQA models with OCR inputs. However, these existing methods are limited by their simple feature representation of image text, multimodal learning approaches, and one-step classification for answer outputs. In this work, we address these limitations with our M4C model.

Multimodal learning in vision-and-language tasks. Early approaches on vision-and-language tasks often combined the image and text through attention over one modality conditioned on the other modality, such as image attention based on text (*e.g.* [51, 34]). Some approaches have explored multimodal fusion mechanisms such as bilinear models (*e.g.* [14, 25]), self-attention (*e.g.* [15]), and graph networks (*e.g.* [30]). Inspired by the success of Transformer [48] and BERT [13] architectures in natural language tasks, several recent works [33, 1, 47, 31, 29, 45, 53, 11] have also applied transformer-based fusion between image and text with self-supervision on large-scale datasets. However, most existing works treat each modality with a specific set of parameters, which makes them hard to scale to more input modalities. On the other hand, in our work we project all entities from each modality into a joint embedding space and treat them homogeneously with a transformer architecture over the list of all things. Our results suggest that joint embedding and self-attention are efficient when modeling multiple (more than two) input modalities.

Dynamic copying with pointers. Many answers in the TextVQA task come from text tokens in the image such as book titles or street signs. As it is intractable to have every possible text token in the answer vocabulary, copying text from the image would often be an easier option for answer prediction. Prior work has explored dynamically copying the inputs in different tasks such as text summarization [42], knowledge retrieval [52], and image captioning [35] based on Pointer Networks [50] and its variants. For the TextVQA task, recent works [44, 37] have proposed to copy OCR tokens by adding their indices to classifier outputs. However, apart from their limitation of copying only a single token (or block), one drawback of these approaches is that they require a pre-defined number of OCR tokens (since the classifier has a fixed output dimension) and their output is dependent on the ordering of the tokens. In this work, we overcome this drawback using a permutation-invariant pointer network together with our multimodal transformer.

3. Multimodal Multi-Copy Mesh (M4C)

In this work, we present Multimodal Multi-Copy Mesh (M4C), a novel approach for the TextVQA task based on a pointer-augmented multimodal transformer architecture with iterative answer prediction. Given a question and an image as inputs, we extract feature representations from three modalities – the question, the visual objects in the image, and the text present in the image. These three modalities are represented respectively as a list of question words features, a list of visual object features from an off-the-shelf object detector, and a list of OCR token features based on an external OCR system.

Our model projects the feature representations of entities (in our case, question words, detected objects, and detected OCR tokens) from the three modalities as vectors in a learned common embedding space. Then, a multi-layer transformer [48] is applied on the list of all projected features, enriching their representations with intra- and inter-modality context. Our model learns to predict the answer through iterative decoding accompanied by a dynamic pointer network. During decoding, it feeds in the previous output to predict the next answer component in an auto-regressive manner. At each step, it either copies an OCR token from the image, or selects a word from its fixed answer vocabulary. Figure 2 shows an overview of our model.

3.1. A common embedding space for all modalities

Our model receives inputs from three modalities – question words, visual objects, and OCR tokens. We extract feature representations for each modality and project them into a common d -dimensional semantic space through domain-specific embedding approaches as follows.

Embedding of question words. Given a question as a sequence of K words, we embed these words into the corresponding sequence of d -dimensional feature vectors $\{x_k^{\text{ques}}\}$ (where $k = 1, \dots, K$) using a pretrained BERT model [13].¹ During training, the BERT parameters are fine-tuned using the question answering loss.

Embedding of detected objects. Given an image, we obtain a set of M visual objects through a pretrained detector (Faster R-CNN [41] in our case). Following prior work [3, 43, 44], we extract appearance feature x_m^{fr} using the detector’s output from the m -th object (where $m = 1, \dots, M$). To capture its location in the image, we introduce a 4-dimensional location feature x_m^b from m -th object’s relative bounding box coordinates $[x_{\min}/W_{\text{im}}, y_{\min}/H_{\text{im}}, x_{\max}/W_{\text{im}}, y_{\max}/H_{\text{im}}]$, where W_{im} and H_{im} are image width and height respectively. Then, the appearance feature and the location feature are

¹In our implementation, we extract question word features from the first 3 layers of BERT-BASE. We find it sufficient to use its first few layers instead of using all its 12 layers, which saves computation.

projected into the d -dimensional space with two learned linear transforms (where d is the same as in the question word embedding above), and are summed up as the final object embedding $\{x_m^{\text{obj}}\}$ as

$$x_m^{\text{obj}} = \text{LN}(W_1 x_m^{\text{fr}}) + \text{LN}(W_2 x_m^b) \quad (1)$$

where W_1 and W_2 are learned projection matrices. $\text{LN}(\cdot)$ is layer normalization [5], added on the output of the linear transforms to ensure that the object embedding has the same scale as the question word embedding. We fine-tune the last layer of the Faster R-CNN detector during training.

Embedding of OCR tokens with rich representations. Intuitively, to represent text in images, one needs to encode not only its characters, but also its appearance (*e.g.* color, font, and background) and spatial location in the image (*e.g.* words appearing on the top of a book cover are more likely to be book titles). We follow this intuition in our model and use a rich OCR representation consisting of four types of features, which is shown in our experiments to be significantly better than word embedding (such as FastText) alone in prior work [44]. After obtaining a set of N OCR tokens in an image through external OCR systems, from the n -th token (where $n = 1, \dots, N$) we extract 1) a 300-dimensional FastText [9] vector x_n^{ft} , which is a word embedding with sub-word information, 2) an appearance feature x_n^{fr} from the same Faster R-CNN detector in the object detection above, extracted via ROI-Pooling on the OCR token’s bounding box, 3) a 604-dimensional Pyramidal Histogram of Characters (PHOC) [2] vector x_n^p , capturing what characters are present in the token – this is more robust to OCR errors and can be seen as a coarse character model, and 4) a 4-dimensional location feature x_n^b based on the OCR token’s relative bounding box coordinates $[x_{\min}/W_{\text{im}}, y_{\min}/H_{\text{im}}, x_{\max}/W_{\text{im}}, y_{\max}/H_{\text{im}}]$. We linearly project each feature into d -dimensional space, and sum them up (after layer normalization) as the final OCR token embedding $\{x_n^{\text{ocr}}\}$ as below

$$x_n^{\text{ocr}} = \text{LN}(W_3 x_n^{\text{ft}} + W_4 x_n^{\text{fr}} + W_5 x_n^p) + \text{LN}(W_6 x_n^b) \quad (2)$$

where W_3, W_4, W_5 and W_6 are learned projection matrices and $\text{LN}(\cdot)$ is layer normalization.

3.2. Multimodal fusion and iterative answer prediction with pointer-augmented transformers

After embedding all entities (question words, visual objects, and OCR tokens) from each modality as vectors in the d -dimensional joint embedding space as described in Sec. 3.1, we apply a stack of L transformer layers [48] with a hidden dimension of d over the list of all $K + M + N$ entities from $\{x_k^{\text{ques}}\}$, $\{x_m^{\text{obj}}\}$, and $\{x_n^{\text{ocr}}\}$. Through the multi-head self-attention mechanism in transformers, each entity is allowed to freely attend to all other entities, regardless of

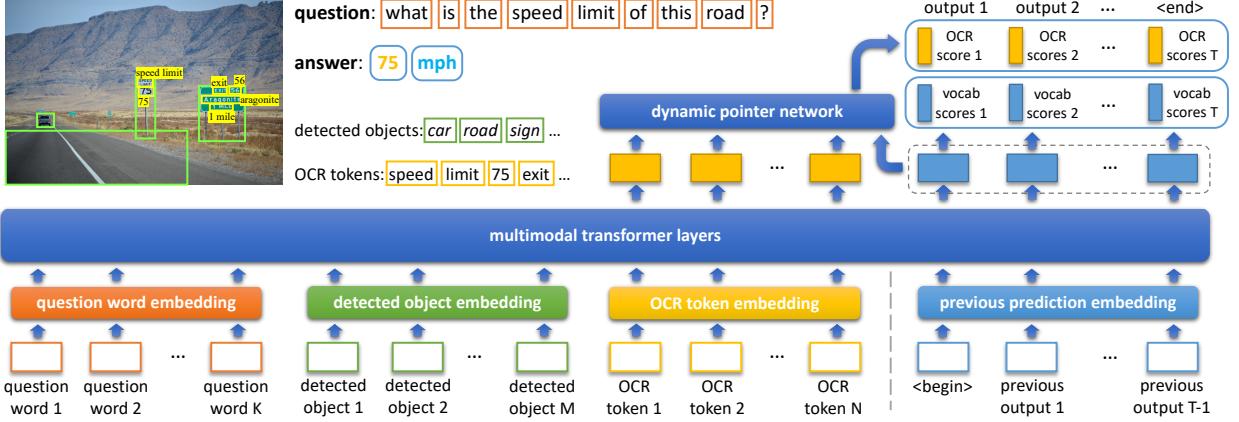


Figure 2. An overview of our M4C model. We project all entities (question words, detected visual objects, and detected OCR tokens) into a common d -dimensional semantic space through domain-specific embedding approaches and apply multiple transformer layers over the list of projected things. Based on the transformer outputs, we predict the answer through iterative auto-regressive decoding, where at each step our model either selects an OCR token through our dynamic pointer network, or a word from its fixed answer vocabulary.

whether they are from the same modality or not. For example, an OCR token is allowed to attend to another OCR token, a detected object, or a question word. This enables modeling both inter- and intra- modality relations in a homogeneous way through the same set of transformer parameters. The output from our multimodal transformer is a list of d -dimensional feature vectors for entities in each modality, which can be seen as their enriched embedding in multimodal context.

We predict an answer to the question through iterative decoding, using exactly the same transformer layers as a decoder. We decode the answer word by word in an auto-regressive manner for a total of T steps, where each decoded word may be either an OCR token in the image or a word from our fixed vocabulary of frequent answer words. As illustrated in Figure 2, at each step during decoding, we feed in an embedding of the previously predicted word, and predict the next answer word based on the transformer output with a dynamic pointer network.

Let $\{z_1^{\text{ocr}}, \dots, z_N^{\text{ocr}}\}$ be the d -dimensional transformer outputs of the N OCR tokens in the image. Assume we have a vocabulary of V words that frequently appear in the training set answers. At the t -th decoding step, the transformer model outputs a d -dimensional vector z_t^{dec} corresponding to the input x_t^{dec} at step t (explained later in this section). From z_t^{dec} , we predict both the V -dimensional scores y_t^{voc} of choosing a word from fixed answer vocabulary and the N -dimensional scores y_t^{ocr} of selecting an OCR token from the image at decoding step t . In our implementation, the fixed answer vocabulary score $y_{t,i}^{\text{voc}}$ for the i -th word (where $i = 1, \dots, V$) is predicted as a simple linear layer as

$$y_{t,i}^{\text{voc}} = (w_i^{\text{voc}})^T z_t^{\text{dec}} + b_i^{\text{voc}} \quad (3)$$

where w_i^{voc} is a d -dimensional parameter for the i -th word in the answer vocabulary, and b_i^{voc} is a scalar parameter.

To select a token from the N OCR tokens in the im-

age, we augment the transformer model with a dynamic pointer network, predicting a copying score $y_{t,n}^{\text{ocr}}$ (where $n = 1, \dots, N$) for each token via bilinear interaction between the decoding output z_t^{dec} and each OCR token's output representation z_n^{ocr} as

$$y_{t,n}^{\text{ocr}} = (W^{\text{ocr}} z_n^{\text{ocr}} + b^{\text{ocr}})^T (W^{\text{dec}} z_t^{\text{dec}} + b^{\text{dec}}) \quad (4)$$

where W^{ocr} and W^{dec} are $d \times d$ matrices, and b^{ocr} and b^{dec} are d -dimensional vectors.

During prediction, we take the argmax on the concatenation $y_t^{\text{all}} = [y_t^{\text{voc}}; y_t^{\text{ocr}}]$ of fixed answer vocabulary scores and dynamic OCR-copying scores, selecting the top scoring element (either a vocabulary word or an OCR token) from all $V + N$ candidates.

In our iterative auto-regressive decoding procedure, if the prediction at decoding time-step t is an OCR token, we feed in its OCR representation x_n^{ocr} as the transformer input x_{t+1}^{dec} to the next prediction step $t + 1$. Otherwise (the previous prediction is a word from the fixed answer vocabulary), we feed in its corresponding weight vector w_i^{voc} in Eqn. 3 as the next step's input x_{t+1}^{dec} . In addition, we add two extra d -dimensional vectors as inputs – a positional embedding vector corresponding to step t , and a type embedding vector corresponding to whether the previous prediction is a fixed vocabulary word or an OCR token. Similar to machine translation, we augment our answer vocabulary with two special tokens, <begin> and <end>. Here <begin> is used as the input to the first decoding step, and we stop the decoding process after <end> is predicted.

To ensure causality in answer decoding, we mask the attention weights in the self-attention layers of the transformer architecture [48] such that question words, detected objects and OCR tokens cannot attend to any decoding steps, and all decoding steps can only attend to previous decoding steps in addition to question words, detected objects and OCR tokens. This is similar to prefix LM in [40].

3.3. Training

During training, we supervise our multimodal transformer at each decoding step. Similar to sequence prediction tasks such as machine translation, we use teacher-forcing [28] (*i.e.* using ground-truth inputs to the decoder) to train our multi-step answer decoder, where each ground-truth answer is tokenized into a sequence of words. Given that an answer word can appear in both fixed answer vocabulary and OCR tokens, we apply multi-label sigmoid loss (instead of softmax loss) over the concatenated scores y_t^{all} .

4. Experiments

We evaluate our model on three challenging datasets for the TextVQA task, including the TextVQA dataset [44], the ST-VQA dataset [8], and the OCR-VQA dataset [37]. Our model outperforms previous work by a significant margin on all the three datasets.

4.1. Evaluation on the TextVQA dataset

The TextVQA dataset [44] contains 28,408 images from the Open Images dataset [27], with human-written questions asking to reason about text in the image. Similar to VQAv2 [17], each question in the TextVQA dataset has 10 human annotated answers, and the final accuracy is measured via soft voting of the 10 answers.²

We use $d = 768$ as the dimensionality of the joint embedding space and extract question word features with BERT-BASE using the 768-dimensional outputs from its first three layers, which are fine-tuned during training.

For visual objects, following Pythia [43] and LoRRA [44], we detect objects with a Faster R-CNN detector [41] pretrained on the Visual Genome dataset [26], and keeps 100 top-scoring objects per image. Then, the fc6 feature vector is extracted from each detected object. We apply the Faster R-CNN fc7 weights on the extracted fc6 features to output 2048-dimensional fc7 appearance features and fine-tune fc7 weights during training. However, we do not use the ResNet-152 convolutional features [19] as in LoRRA.

Finally, we extract text tokens on each image using the Rosetta OCR system [10]. Unlike the prior work LoRRA [44] that uses a multilingual Rosetta version, in our model we use an English-only version of Rosetta that we find has higher recall. We refer to these two versions as **Rosetta-ml** and **Rosetta-en**, respectively. As mentioned in Sec. 3.1, from each OCR token we extract **FastText** [9] feature, appearance feature from Faster R-CNN (**FRCN**), **PHOC** [2] feature, and bounding box (**bbox**) feature.

In our multimodal transformer, we use $L = 4$ layers of multimodal transformer with 12 attention heads. Other hyper-parameters (such as dropout ratio) follow BERT-BASE [13]. However, we note that the multimodal trans-

former parameters are initialized from scratch rather than from a pretrained BERT model. We use $T = 12$ maximum decoding step in answer prediction unless otherwise specified, which is sufficient to cover almost all answers.

We collect the top 5000 frequent words from the answers in the training set as our answer vocabulary. During training, we use a batch size of 128, and train for a maximum of 24,000 iterations. Our model is trained using the Adam optimizer, with a learning rate of 1e-4 and a staircase learning rate schedule, where we multiply the learning rate by 0.1 at 14000 and at 19000 iterations. The best snapshot is selected using the validation set accuracy. The entire training takes approximately 10 hours on 4 Nvidia Tesla V100 GPUs.

As a notable prior work on this dataset, we show a step-by-step comparison with the LoRRA model [44]. LoRRA uses two single-hop attention layers over image visual features and OCR features. The attended visual and OCR features are then fused with a vector encoding of the question and fed into a single-step classifier to select either a frequent answer from the training set or a single OCR token from the image. Unlike our rich OCR representation in Sec. 3.1, in the LoRRA model each OCR token is only represented as a 300-dimensional FastText vector.

Ablations on pretrained question encoding and OCR systems. We first experiment with a restricted version of our model using the multimodal transformer architecture but without iterative decoding in answer prediction, *i.e.* **M4C (w/o dec.)** in Table 1. In this setting, we only decode for one step, and either select a frequent answer³ from the training set or copy a single OCR token in the image as the answer. As a step-by-step comparison with LoRRA, we start with extracting OCR tokens from Rosetta-ml, representing OCR tokens only with FastText vectors, and initializing question encoding parameters in Sec. 3.1 from scratch (rather than from a pretrained BERT-BASE model). The result is shown in line 2 of Table 1. Compared with LoRRA in line 1, this restricted version of our model already outperforms LoRRA by around 3% (absolute) on TextVQA validation set. Given that LoRRA uses pretrained GloVe [39] for question encoding while we learn question encoding from scratch in line 2, this result shows that our multimodal transformer architecture is more efficient for jointly modeling the three input modalities. We then switch to a pretrained BERT for question encoding in line 3, and Rosetta-en for OCR extraction in line 4. Comparing line 2 to 4, we see that a pretrained BERT leads to around 0.6% higher accuracy, and Rosetta-en gives another 1% improvement.

Ablations on OCR feature representation. We analyze the impact of our rich OCR representation in Sec. 3.1 through ablations in Table 1 line 4 to 7. We see that OCR

²See <https://visualqa.org/evaluation> for details.

³In this case, we predict the entire (multi-word) answer, instead of a single word from our answer word vocabulary as in our full model.

#	Method	Question enc. pretraining	OCR system	OCR token representation	Output module	Accu. on val	Accu. on test
1	LoRRA [44]	GloVe	Rosetta-ml	FastText	classifier	26.56	27.63
2	M4C w/o dec.	(none)	Rosetta-ml	FastText	classifier	29.55	—
3	M4C w/o dec.	BERT	Rosetta-ml	FastText	classifier	30.15	—
4	M4C w/o dec.	BERT	Rosetta-en	FastText	classifier	31.28	—
5	M4C w/o dec.	BERT	Rosetta-en	FastText + bbox	classifier	33.32	—
6	M4C w/o dec.	BERT	Rosetta-en	FastText + bbox + FRCN	classifier	34.38	—
7	M4C w/o dec.	BERT	Rosetta-en	FastText + bbox + FRCN + PHOC	classifier	35.70	—
8	M4C (ours - ablation)	(none)	Rosetta-ml	FastText + bbox + FRCN + PHOC	decoder	36.06	—
9	M4C (ours - ablation)	BERT	Rosetta-ml	FastText + bbox + FRCN + PHOC	decoder	37.06	—
10	M4C (ours)	BERT	Rosetta-en	FastText + bbox + FRCN + PHOC	decoder	39.40	39.01
11	DCD_ZJU (ensemble) [32]	—	—	—	—	31.48	31.44
12	MSFT_VTI [46]	—	—	—	—	32.92	32.46
13	M4C (ours; w/ ST-VQA)	BERT	Rosetta-en	FastText + bbox + FRCN + PHOC	decoder	40.55	40.46

Table 1. On the TextVQA dataset, we ablate our M4C model and show a detailed comparison with prior work LoRRA [44]. Our multimodal transformer (line 2 vs 1), our rich OCR representation (line 7 vs 4) and our iterative answer prediction (line 10 vs 7) all improve the accuracy significantly. Notably, our model still outperforms LoRRA by 9.5% (absolute) even when using fewer pretrained parameters (line 8 vs 1). Our final model achieves 39.01% (line 10) and 40.46% (line 13) test accuracy without and with the ST-VQA dataset as additional training data respectively, outperforming the challenge-winning DCD_ZJU method by 9% (absolute). See Sec. 4.1 for details.

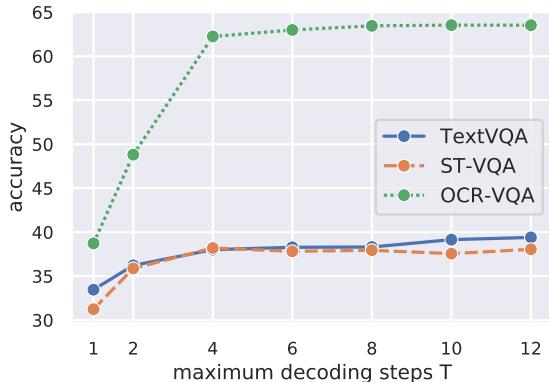


Figure 3. Accuracy under different maximum decoding steps T on the validation set of TextVQA, ST-VQA, and OCR-VQA. There is a major gap between single-step ($T = 1$) and multi-step ($T > 1$) answer prediction. We use 12 steps by default in our experiments.

location (bbox) features and the RoI-pooled appearance features (FRCN) both improve the performance by a noticeable margin. In addition, we find that PHOC is also helpful as a character-level representation of the OCR token. Our rich OCR representation gives around 4% (absolute) accuracy improvement compare with using only FastText features as in LoRRA (line 7 vs 4). We note that our extra OCR features do not require more pretrained models, as we apply exactly the same Faster R-CNN model use in object detection for OCR appearance features, and PHOC is a manually-designed feature that does not need pretraining.

Iterative answer decoding. We then apply our full M4C model with iterative answer decoding to the TextVQA dataset. The results are shown in Table 1 line 10, which is around 4% (absolute) higher than its counterpart in line 7 using a single-step classifier and 13% (absolute) higher

than LoRRA in line 1. In addition, we ablate our model using Rosetta-ml and randomly initialized question encoding parameters in line 8 and 9. Here, we see that our model in line 8 still outperforms LoRRA (line 1) by as much as 9.5% (absolute) when using the same OCR system as LoRRA and even fewer pretrained components. We also analyze the performance of our model with respect to the maximum decoding steps, shown in Figure 3, where decoding for multiple steps greatly improves the performance compared with a single step. Figure 4 shows qualitative examples (more examples in appendix) of our M4C model on the TextVQA dataset in comparison to LoRRA [44], where our model is capable of selecting multiple OCR tokens and combining them with its fixed vocabulary in predicted answers.

Qualitative insights. When inspecting the errors, we find that a major source of errors is OCR failure (*e.g.* in the last example in Figure 4, we find that the digits on the watch are not detected). This suggests that the accuracy of our model could be improved with better OCR systems, as supported by the comparison between line 9 and 10 in Table 1. Another possible future direction is to dynamically recognize text in the image based on the question (*e.g.* if the question asks about the price of a product brand, one may want to directly localize the brand name in the image). Some other errors of our model include resolving relations between objects and text or understanding large chunks of text in images (such as book pages). However, our model is able to correct a large number of mistakes in previous work where copying multiple text tokens is required to form an answer.

TextVQA Challenge 2019. We also compare to the winning entries in the TextVQA Challenge 2019.⁴ We compare

⁴<https://textvqa.org/challenge>

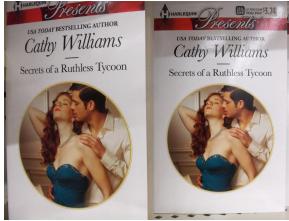


What does the light sign read on the farthest right window?

LoRRA: **exit**

M4C (ours): **bud light**

human: **bud light; all 2 liters**

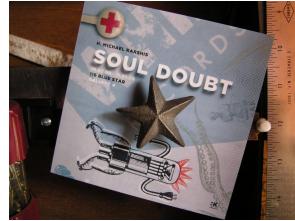


Who is usa today's bestselling author?

LoRRA: **roger zelazny**

M4C (ours): **cathy williams**

human: **cathy williams**



What is the name of the band?

LoRRA: **7**

M4C (ours): **soul doubt**

human: **soul doubt; h. michael karshis; unanswerable**



what is the time?

LoRRA: **1:45**

M4C (ours): **3:44**

human: **5:40; 5:41; 5:42; 8:00**

Figure 4. Qualitative examples from our M4C model on the TextVQA validation set (orange words are from OCR tokens and blue words are from fixed answer vocabulary). Compared to the previous work LoRRA [44] which selects one answer from training set or copies only a single OCR token, our model can copy multiple OCR tokens and combine them with its fixed vocabulary through iterative decoding.

our method to DCD [32] (the challenge winner, based on ensemble) and MSFT_VTI [46] (the top entry after the challenge), both relying on one-step prediction. We show that our single model (line 10) significantly outperforms these challenge winning entries on the TextVQA test set by a large margin. We also experiment with using the ST-VQA dataset [8] as additional training data (a practice used by some of the previous challenge participants), which gives another 1% improvement and 40.46% final test accuracy – a new state-of-the-art on the TextVQA dataset.

4.2. Evaluation on the ST-VQA dataset

The ST-VQA dataset [8] contains natural images from multiple sources including ICDAR 2013 [24], ICDAR 2015 [23], ImageNet [12], VizWiz [18], IIIT STR [36], Visual Genome [26], and COCO-Text [49].⁵ The format of the ST-VQA dataset is similar to the TextVQA dataset in Sec. 4.1. However, each question is accompanied by only one or two ground-truth answers provided by the question writer. The dataset involves three tasks, and its Task 3 - Open Dictionary (containing 18,921 training-validation images and test 2,971 images) corresponds to our general TextVQA setting where no answer candidates are provided at test time.

The ST-VQA dataset adopts Average Normalized Levenshtein Similarity (ANLS)⁶ as its official evaluation metric, defined as scores $1 - d_L(a_{\text{pred}}, a_{\text{gt}}) / \max(|a_{\text{pred}}|, |a_{\text{gt}}|)$ (where a_{pred} and a_{gt} are prediction and ground-truth answers and d_L is edit distance) averaged over all questions. Also, all scores below the threshold 0.5 are truncated to 0 before averaging. To facilitate comparison, we report both accuracy and ANLS in our experiments.

As the ST-VQA dataset does not have an official split for training and validation, we randomly select 17,028 im-

⁵We notice that many images from COCO-Text [49] in the downloaded ST-VQA data (around 1/3 of all images) are resized to 256×256 for unknown reasons, which degrades the image quality and distorts their aspect ratios. In our experiments, we replace these images with their original versions from COCO-Text as inputs to object detection and OCR systems.

⁶<https://rrc.cvc.uab.es/?ch=11&com=tasks>

#	Method	Output module	Accu. on val	ANLS on val	ANLS on test
1	SAN+STR [8]	–	–	–	0.135
2	VTA [7]	–	–	–	0.282
3	M4C w/o dec.	classifier	33.52	0.397	–
4	M4C (ours)	decoder	38.05	0.472	0.462

Table 2. On the ST-VQA dataset, our restricted model without decoder (M4C w/o dec.) already outperforms previous work by a large margin. Our final model achieves +0.18 (absolute) ANLS boost over the challenge winner, VTA [7]. See Sec. 4.2 for details.

ages as our training set and use the remaining 1,893 images as our validation set. We train our model on the ST-VQA dataset following exactly the same setting (line 10 in Table 1) as in our TextVQA experiments in Sec. 4.1, where we extract image text tokens using Rosetta-en, use FastText + bbox + FRCN + PHOC as our OCR representation, and initialize question encoding parameters from a pretrained BERT-BASE model. The results are shown in Table 2.

Ablations of our model. We train two versions of our model, one restricted version (M4C w/o dec. in Table 2) with a fixed one-step classifier as output module (similar to line 7 in Table 1) and one full version (M4C) with iterative answer decoding. Comparing the results of these two models, it can be seen that there is a large improvement from our iterative answer prediction mechanism.

Comparison to previous work. We compare with two previous methods on this dataset: 1) SAN+STR [8], which combines SAN for VQA [51] and Scene Text Retrieval [16] for answer vocabulary retrieval, and 2) VTA [7], the ICDAR 2019 ST-VQA Challenge⁶ winner, based on BERT [13] for question encoding and BUTD [3] for VQA. From Table 2, it can be seen that our restricted model (M4C w/o dec.) already achieves higher ANLS than these two models, and our full model achieves as much as +0.18 (absolute) ANLS boost over the best previous work.

We also ablate the maximum copying number in our



What is the name of the street on which the Stop sign appears?
prediction: **45th parallel dr**
GT: **45th parallel dr**



What does the white sign say?
prediction: **tokyo station**
GT: **tokyo station**



How many cents per pound are the bananas?
prediction: **99**
GT: **99**

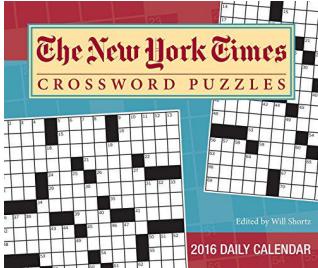


What kind of stop sign is in the image?
prediction: **stop all way**
GT: **all way**

Figure 5. Qualitative examples from our M4C model on the ST-VQA validation set (**orange** words from OCR tokens and **blue** words from fixed answer vocabulary). Our model can select multiple OCR tokens and combine them with its fixed vocabulary to predict an answer.

#	Method	Output module	Accu. on val	Accu. on test
1	BLOCK [37]	–	–	42.0
2	CNN [37]	–	–	14.3
3	BLOCK+CNN [37]	–	–	41.5
4	BLOCK+CNN+W2V [37]	–	–	48.3
5	M4C w/o dec.	classifier	46.3	–
6	M4C (ours)	decoder	63.5	63.9

Table 3. On the OCR-VQA dataset, we experiment with using either an iterative decoder (our full model) or a single-step classifier (M4C w/o dec.) as the output module, where our iterative decoder greatly improves the accuracy and largely outperforms the baseline methods. See Sec. 4.3 for details.



Who is the author of this book?
prediction: **the new york times**
GT: **the new york times**



Is this a pharmaceutical book?
prediction: **no**
GT: **no**

Figure 6. Qualitative examples from our M4C model on the OCR-VQA validation set (**orange** words from OCR tokens and **blue** words from fixed answer vocabulary).

model in Figure 3, showing that it is beneficial to decode for multiple (as opposed to one) steps. Figure 5 shows qualitative examples of our model on the ST-VQA dataset.

4.3. Evaluation on the OCR-VQA dataset

The OCR-VQA dataset [37] contains 207,572 images of book covers, with template-based questions asking about the title, author, edition, genre, year or other information about the book. Each question has a single ground-truth answer, and the dataset assumes that the answers to these

questions can be inferred from the book cover images.

We train our model using the same hyper-parameters as in Sec. 4.1 and 4.2, but use $2\times$ the total iterations and adapted learning rate schedule since the OCR-VQA dataset contains more images. The results are shown in Table 3. Compared to using a one-step classifier (M4C w/o dec.), our full model with iterative decoding achieves significantly better accuracy, which coincides with Figure 3 that having multiple decoding steps is greatly beneficial on this dataset. This is likely because the OCR-VQA dataset often contains multi-word answers such as book titles and author names.

We compare to four baseline approaches from [37], which are VQA systems based on 1) visual features from a convolutional network (CNN), 2) grouping OCR tokens into text blocks (BLOCK) with manually defined rules, 3) an averaged word2vec (W2V) feature over all the OCR tokens in the image, and 4) their combinations. Note that while the BLOCK baseline can also select multiple OCR tokens, it relies on manually defined rules to merge tokens into groups and can only select one group as answer, while our method learns from data how to copy OCR tokens to compose answers. Compare to these baselines, our M4C has over 15% (absolute) higher test accuracy. Figure 6 shows qualitative examples of our model on this dataset.

5. Conclusion

In this paper, we present Multimodal Multi-Copy Mesh (M4C) for visual question answering based on understanding and reasoning about text in images. M4C adopts rich representations for text in the images, jointly models all modalities through a pointer-augmented multimodal transformer architecture over a joint embedding space, and predicts the answer through iterative decoding, outperforming previous work by a large margin on three challenging datasets for the TextVQA task. Our results suggest that it is efficient to handle multiple modalities through domain-specific embedding followed by homogeneous self-attention and to generate complex answers as multi-step decoding instead of one-step classification.

References

- [1] Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. Fusion of detected objects in text for visual question answering. *arXiv preprint arXiv:1908.05054*, 2019. 2
- [2] Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny. Word spotting and recognition with embedded attributes. *IEEE transactions on pattern analysis and machine intelligence*, 36(12):2552–2566, 2014. 3, 5
- [3] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018. 1, 3, 7
- [4] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015. 1
- [5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3
- [6] Hedi Ben-Younes, Rémi Cadene, Matthieu Cord, and Nicolas Thome. Mutan: Multimodal tucker fusion for visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2612–2620, 2017. 1
- [7] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusiñol, Minesh Mathew, CV Jawahar, Ernest Valveny, and Dimosthenis Karatzas. Icdar 2019 competition on scene text visual question answering. *arXiv preprint arXiv:1907.00490*, 2016. 1, 2, 7
- [8] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusiñol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 1, 2, 5, 7
- [9] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. 3, 5
- [10] Fedor Borisyuk, Albert Gordo, and Viswanath 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. 5
- [11] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019. 2
- [12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. 2, 3, 5, 7
- [14] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 457–468, 2016. 1, 2
- [15] Peng Gao, Zhengkai Jiang, Haoxuan You, Pan Lu, Steven CH Hoi, Xiaogang Wang, and Hongsheng Li. Dynamic fusion with intra-and inter-modality attention flow for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6639–6648, 2019. 2
- [16] Lluís Gómez, Andrés Mafla, Marçal Rusinol, and Dimosthenis Karatzas. Single shot scene text retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 700–715, 2018. 7
- [17] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017. 1, 5
- [18] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3608–3617, 2018. 7
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5
- [20] Drew A Hudson and Christopher D Manning. Gqa: a new dataset for compositional question answering over real-world images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1
- [21] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2910, 2017. 1
- [22] Kushal Kafle and Christopher Kanan. An analysis of visual question answering algorithms. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1965–1973, 2017. 1
- [23] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Angelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE, 2015. 7

- [24] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493. IEEE, 2013. 7
- [25] Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, pages 1564–1574, 2018. 1, 2
- [26] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. 5, 7
- [27] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Tom Duerig, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018. 5
- [28] Alex M Lamb, Anirudh Goyal Alias Parth Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pages 4601–4609, 2016. 5
- [29] Gen Li, Nan Duan, Yuejian Fang, Dixin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*, 2019. 2
- [30] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. *arXiv preprint arXiv:1903.12314*, 2019. 2
- [31] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 2
- [32] Yuetan Lin, Hongrui Zhao, Yanan Li, and Donghui Wang. DCD_ZJU, TextVQA Challenge 2019 winner. <https://visualqa.org/workshop.html>. 6, 7
- [33] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, 2019. 1, 2
- [34] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems*, pages 289–297, 2016. 2
- [35] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7219–7228, 2018. 2
- [36] Anand Mishra, Karteek Alahari, and CV Jawahar. Image retrieval using textual cues. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3040–3047, 2013. 7
- [37] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *Proceedings of the International Conference on Document Analysis and Recognition*, 2019. 1, 2, 5, 8
- [38] Ankur P Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*, 2016. 2
- [39] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 5
- [40] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 4
- [41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3, 5
- [42] Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*, 2017. 2
- [43] Amanpreet Singh, Vivek Natarajan, Yu Jiang, Xinlei Chen, Meet Shah, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia-a platform for vision & language research. In *SysML Workshop, NeurIPS*, volume 2018, 2018. 1, 2, 3, 5
- [44] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019. 1, 2, 3, 5, 6, 7, 12
- [45] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*, 2019. 2
- [46] Anonymous submission. MSFT_VTI, TextVQA Challenge 2019 top entry (post-challenge). <https://evalai.cloudcv.org/web/challenges/challenge-page/244/>. 6, 7
- [47] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019. 2
- [48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 2, 3, 4
- [49] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016. 7
- [50] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700, 2015. 2

- [51] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016. [2](#), [7](#)
- [52] Semih Yavuz, Abhinav Rastogi, Guan-Lin Chao, and Dilek Hakkani-Tur. Deepcopy: Grounded response generation with hierarchical pointer networks. *arXiv preprint arXiv:1908.10731*, 2019. [2](#)
- [53] Luwei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. *arXiv preprint arXiv:1909.11059*, 2019. [2](#)

Iterative Answer Prediction with Pointer-Augmented Multimodal Transformers for TextVQA

(Supplementary Material)

A. Hyper-parameters in M4C

We summarize the hyper-parameters in our M4C model in Table 4. Most hyper-parameters are the same across all the three datasets (TextVQA, ST-VQA, and OCR-VQA), except that we use $2\times$ the total iterations and adapted learning rate schedule on the OCR-VQA dataset since it contains more images.

Hyper-parameter	Value
max question word num K	20
detected object num M	100
max OCR num N	50
max decoding steps T	12
embedding dim d	768
multimodal transformer layers L	4
multimodal transformer attention heads	12
multimodal transformer FFN dim	3072
multimodal transformer dropout	0.1
optimizer	Adam
batch size	128
base learning rate	1e-4
warm-up learning rate factor	0.2
warm-up iterations	2000
max gradient L2-norm for clipping	0.25
learning rate decay	0.1
learning rate steps (TextVQA, ST-VQA)	14000, 19000
learning rate steps (OCR-VQA)	28000, 38000
max iterations (TextVQA, ST-VQA)	24000
max iterations (OCR-VQA)	48000

Table 4. Hyper-parameters of our M4C.

B. Additional ablation analysis

During the iterative answer decoding process, at each step our M4C model can decode an answer word either from the model’s fixed vocabulary, or from the OCR tokens extracted from the image. We find in our experiments that it is necessary to have *both* the fixed vocabulary space and the OCR tokens.

Table 5 shows our ablation study where we remove the fixed answer vocabulary or the dynamic pointer network for OCR copying from our M4C. Both these two ablated versions have a large accuracy drop compared to our full

model. However, we note that even without fixed answer vocabulary, our restricted model (**M4C w/o fixed vocabulary** in Table 5) still outperforms the previous work LoRRA [44], suggesting that it is particularly important to learn to copy multiple OCR tokens to form an answer (a key feature in our model but not in LoRRA).

#	Method	TextVQA Val Accuracy
1	LoRRA [44]	26.56
2	M4C w/o fixed vocabulary	31.76
3	M4C w/o OCR copying	14.94
4	M4C (ours)	39.40

Table 5. We ablate our M4C model by removing its fixed answer vocabulary (M4C w/o fixed vocabulary) or its dynamic pointer network for OCR copying (M4C w/o OCR copying) on the TextVQA dataset. We see that our full model has significantly higher accuracy than these ablations, showing that it is important to have *both* a fixed and a dynamic vocabulary (*i.e.* OCR tokens).

C. Additional qualitative examples

As mentioned in Sec. 4.1 in the main paper, we find that OCR failure is a major source of error for our M4C model’s predictions. Figure 7 shows cases on the TextVQA dataset where the OCR system fails to precisely localize the corresponding text tokens in the image, suggesting that our model’s accuracy can be improved with better OCR systems.

Figure 8, 9, and 10 shows additional qualitative examples from our M4C model on the TextVQA dataset, ST-VQA, and OCR-VQA datasets, respectively. While our model occasionally fails when reading a large piece of text or resolving the relation between text and objects as in Figure 8 (f) and (h), in most cases it learns to identify and copy text tokens from the image and combine them with its fixed vocabulary to predict an answer.



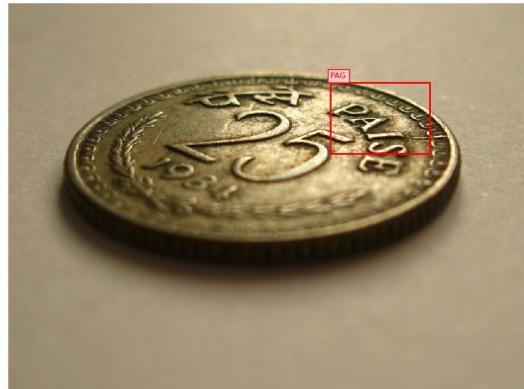
(a) what candy bar is down there on the bottom?
prediction: **unanswerable**
human: **hershey's; hersheys**



(b) what is the year on the calender?
prediction: **2005**
human: **2010; unanswerable**



(c) what is the largest measurement we can see on this ruler?
prediction: **40**
human: **50**



(d) how much is the coin worth?
prediction: **one dollar**
human: **20; 25; 25 paise**



(e) what is the name of the bar?
prediction: **15**
human: **moo bar; moon; moon bar**



(f) what time is it?
prediction: **76**
human: **13:50; 13:57; ;5713; mathematic; wifi**

Figure 7. Examples where OCR failure is the main source of errors (from our M4C model on the TextVQA validation set). The red boxes show the OCR results (**orange** words from OCR tokens and **blue** words from fixed answer vocabulary).



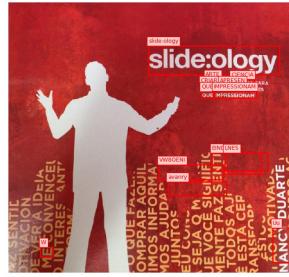
(a) what is the brand of this camera?

M4C: **dakota digital**
human: **dakota digital; dakota; clos culombu; nous les gosses**



(b) does it say happy birthday?

M4C: **yes**
human: **yes**



(c) what is the title of the album?

M4C: **slide:ology**
human: **slide:ology; sideology**



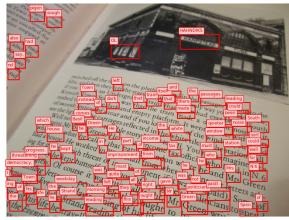
(d) what is the 4 digit number written at the bottom of the black book?

M4C: **9350**
human: **9350; 9,350**



(e) what airline is the plane from?

M4C: **lufthansa**
human: **lufthansa**



(f) what was mr. green's first name?

M4C: **charles**
human: **basil**



(g) what time is displayed on the phone's screen?

M4C: **9:09**
human: **9:09; no**



(h) what number is on the bike on the right?

M4C: **30**
human: **317**



(a) What is this building used for according to the sign above it?

M4C: **post office**
GT: **post office**



(b) What can you get 6 of for \$5?

M4C: **donuts**
GT: **donuts**



(c) where can I buy shoes here?

M4C: **public market**
GT: **footaction**



(d) What is the license plate number on the red car?

M4C: **gsv 820**
GT: **gsv 820**



(e) What does the large pink text say?

M4C: **me**
GT: **pardon me prime minister**



(f) What brand of typewriter is being used?

M4C: **olympia**
GT: **olympia**



(g) What 4-digit number is on the yellow stick in front of the green car?

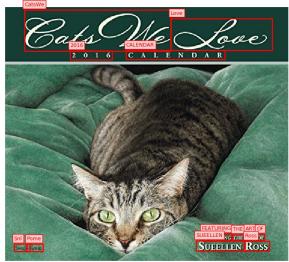
M4C: **4764**
GT: **4764**



(h) What brand is the bike in front?

M4C: **ducati**
GT: **ducati**

Figure 8. Additional qualitative examples from our M4C model on the TextVQA validation set. The red boxes show the OCR results (best viewed in 400%; orange words from OCR tokens and blue words from fixed answer vocabulary).



(a) Who is the author of this book?
M4C: **sueellen ross**

GT: sueellen ross



(b) Which year's calendar is this?
M4C: **2016**

GT: **2016**



(c) What is the title of this book?
M4C: **sailing to the mark 2013 calendar**

GT: **sailing to the mark 2013 calendar**



(d) What is the genre of this book?
M4C: **arts & photography**

GT: **calendars**

Figure 10. Additional qualitative examples from our M4C model on the OCR-VQA validation set. The red boxes show the OCR results (best viewed in 400%; **orange** words from OCR tokens and **blue** words from fixed answer vocabulary).