Simple is not Easy: A Simple Strong Baseline for TextVQA and TextCaps

Qi Zhu,[1] **Chenyu Gao,**[1] **Peng Wang***,[1] **Qi Wu**[2]
[1] Northwestern Polytechnical University, China
[2] University of Adelaide, Australia
{zhu_qi_happy_,chenyugao}@mail.nwpu.edu.cn, peng.wang@nwpu.edu.cn, qi.wu01@adelaide.edu.au

Abstract

Texts appearing in daily scenes that can be recognized by OCR (Optical Character Recognition) tools contain significant information, such as street name, product brand and prices. Two tasks – text-based visual question answering and text-based image captioning, with a text extension from existing vision-language applications, are catching on rapidly. To address these problems, many sophisticated multi-modality encoding frameworks (such as heterogeneous graph structure) are being used. In this paper, we argue that a simple attention mechanism can do the same or even better job without any bells and whistles. Under this mechanism, we simply split OCR token features into separate visual- and linguistic-attention branches, and send them to a popular Transformer decoder to generate answers or captions. Surprisingly, we find this simple baseline model is rather strong – it consistently outperforms state-of-the-art (SOTA) models on two popular benchmarks, TextVQA and all three tasks of ST-VQA, although these SOTA models use far more complex encoding mechanisms. Transferring it to text-based image captioning, we also surpass the TextCaps Challenge 2020 winner. We wish this work to set the new baseline for these two OCR text related applications and to inspire new thinking of multi-modality encoder design. Code is available at https://github.com/ZephyrZhuQi/ssbaseline

Introduction

To automatically answer a question or generate a description for images that require scene text understanding and reasoning has broad prospects for commercial applications, such as assisted driving and online shopping. Equipped with these abilities, a model can help drivers decide distance to the next street or help customers get more details about a product. Two kinds of tasks that focus on text in images have recently been introduced, which are text-based visual question answering (TextVQA) (Singh et al. 2019; Biten et al. 2019) and text-based image captioning (TextCaps) (Sidorov et al. 2020). For example, in Figure 1, a model is required to answer a question or generate a description by reading and reasoning the texts "tellus mater inc." in the image. These two tasks pose a challenge to current VQA or image captioning models as they explicitly require understanding of a new modality – Optical Character Recognition (OCR). A model must efficiently utilize text-related features to solve these problems.

Figure 1: An example of TextVQA and TextCaps tasks. The answer and description are generated by our model. Our simple baseline is able to read texts and answer related questions. Besides, it can also observe the image and generate a description with texts embedded in it.

For TextVQA task, the current state-of-the-art model M4C (Hu et al. 2019) handles all modalities (questions, visual objects and OCR tokens) over a joint embedding space. Although this homogeneous method is easy to implement, fast to train and has made great headway, it considers that texts and visual objects contribute indiscriminately to this problem and uses text features as a whole. For TextCaps problem, the only difference is that it only has two modalities: visual objects and OCR tokens. However, these limitations remain.

Some other works proposed even more complex structures to encode and fuse multi-modality features of this task, *i.e.,* questions, OCR tokens and images. For example,

*Peng Wang is corresponding author.

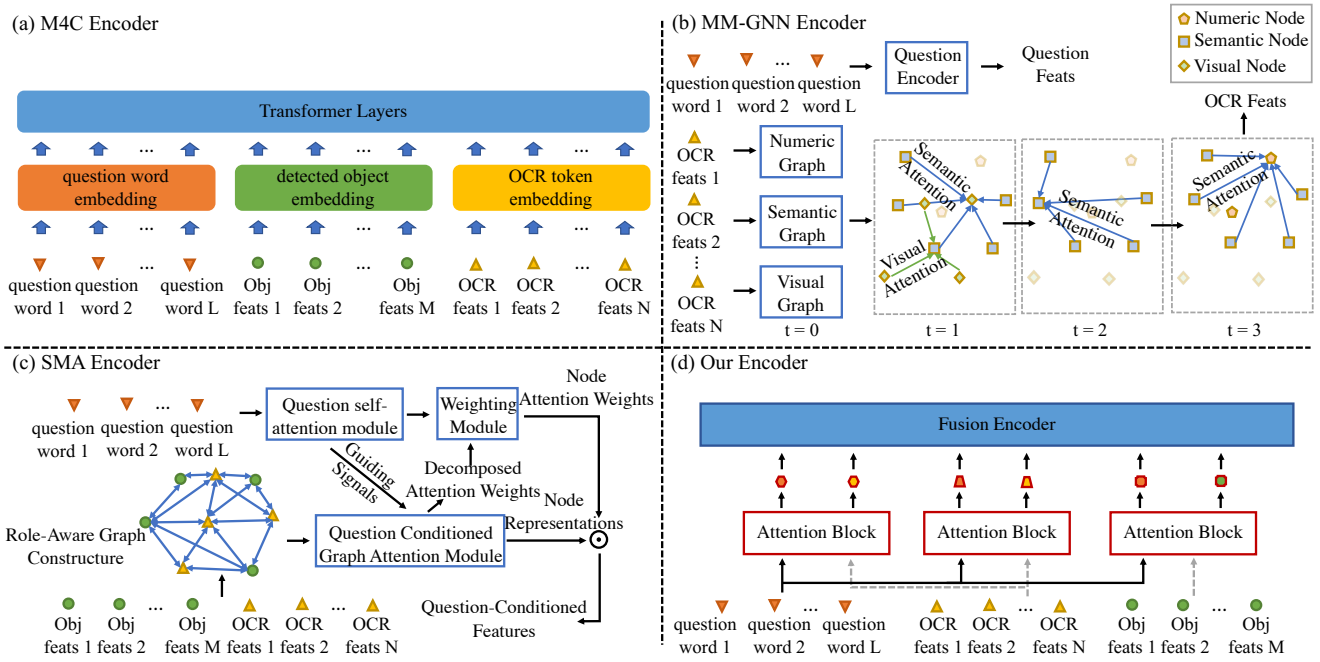

Figure 2: Encoders of different models. (a) Current state-of-the-art model M4C on TextVQA task forwards each feature vector of all modalities indiscriminately into transformer layers, which exhaust tremendous computation. (b) MM-GNN handcrafts three graphs to represent the image and applies three aggregators step by step to pass messages between graphs. (c) SMA introduces a heterogeneous graph and considers object-object, object-text and text-text relationships, upon which a graph attention network is then used to reason over them. (d) Our baseline uses three vanilla attention blocks to highlight most relevant features and combines them into six individually-functioned vectors, which is then sent into transformer-based fusion encoders. The considerably fewer parameters of six vectors save computation.

SMA (Gao et al. 2020b) uses a heterogeneous graph to encode object-object, object-text and text-text relationships in the image, and then designs a graph attention network to reason over it. MM-GNN (Gao et al. 2020c) represents an image as three graphs and introduces three aggregators to guide message passing from one graph to another.

In this paper, we use the vanilla attention mechanism to fuse pairwise modalities. Under this mechanism, we further use a more reasonable method to utilize text features which leads to a higher performance, that is by splitting text features into two functionally different parts, *i.e.,* linguistic- and visual-part which flows into corresponding attention branch. The encoded features are then sent to a popularly-used Transformer-based decoder to generate answers or captions. As compared to the aforementioned M4C models (shown in Figure 2a) that throw each instance of every modality into transformer layers, our model (in Figure 2d) first uses three attention blocks to filter out irrelevant or redundant features and aggregate them into six individually-functioned vectors. In contrast to hundreds of feature vectors in M4C, the six vectors consume much less computation. Moreover, to group text features into visual- and linguistic- parts is more reasonable. When comparing with the graph-based multi-modal encoders such as MM-GNN (Figure 2b) and SMA (Figure 2c), our baseline is extremely simple in design and reduces much space and time complexity.

In addition, for the first time we ask the question: to what extent OCRs contribute to the final performance of TextVQA, in contrast to the other modality - visual contents such as objects and scenes? An interesting phenomenon is observed that OCRs play an almost key role in this special problem while visual contents only serve as assisting factors. A strong model without the use of visual contents surpasses current state-of-the-art model, demonstrating the power of proposed pairwise fusion mechanism and primary role of texts.

To demonstrate the effectiveness of our proposed simple baseline model, we test it on both TextVQA (Singh et al. 2019) and TextCaps (Sidorov et al. 2020) tasks. For the TextVQA, we outperforms the state-of-the-art (SOTA) on TextVQA dataset and all three tasks of ST-VQA, and rank the first on both leaderboards. More importantly, all compared SOTA models use the similar Transformer decoder with ours, but with far more complex encoding mechanisms. For TextCaps, we surpass the TextCaps Challenge 2020 winner and now rank the first place on the leaderboard.

Overall, the major contribution of this work is to provide a simple but rather strong baseline for the text-based vision-and-language research. This could be the new baseline (backbone) model for both TextVQA and TextCaps. More importantly, we wish this work to inspire a new thinking of multi-modality encoder design – simple is not easy.

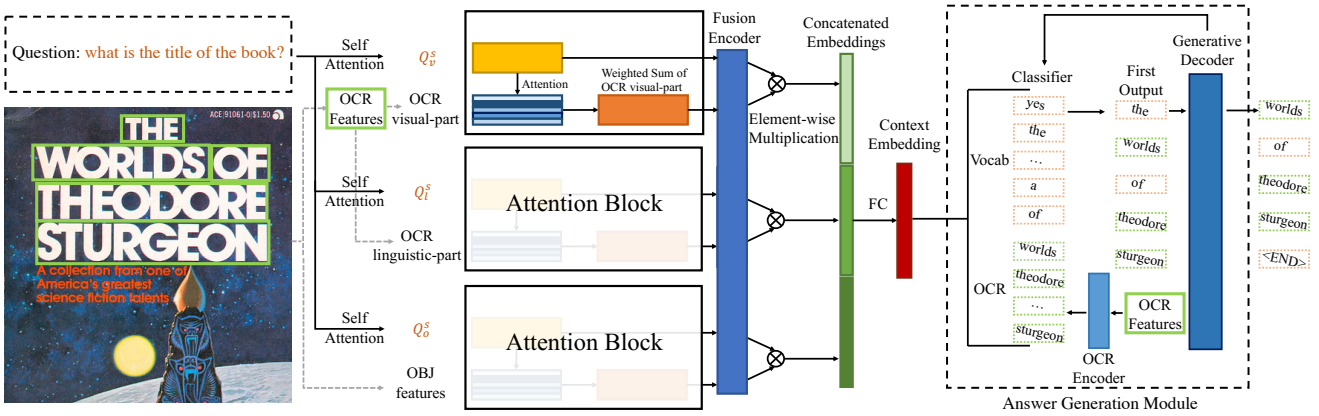

Figure 3: A simple baseline model for TextVQA. Given an image and a question, we prepare three features (OCR visual-part, OCR linguistic-part and object features) and three question self-attention outputs. The six sequences are put into attention block and fused into six vectors, upon which we calculate element-wise product two by two to get concatenated embeddings. The encoder outputs predict the first word and the rest of answer is produced by an iterative decoder.

Related Work

Text based visual question answering. VQA (Antol et al. 2015; Johnson et al. 2017; Kahou et al. 2018; Wang et al. 2018) has seen rapid development in recent years. A new task – TextVQA goes one step further and aims at the understanding and reasoning of scene texts in images. A model needs to read texts first and then answer related questions in natural everyday situations. Two datasets, TextVQA (Singh et al. 2019) and ST-VQA (Biten et al. 2019) are introduced concurrently to benchmark progress in this field. To solve this problem, various methods have also been proposed. LoRRA, the baseline model in TextVQA, uses bottom-up and top-down (Anderson et al. 2018) attention on visual objects and texts to select an answer from either vocabulary or fixed-index OCR. M4C (Hu et al. 2019) is equipped with a vanilla transformer decoder to iteratively generate answers and a flexible pointer network to point back at most probable OCR token at one decoding step. MM-GNN (Gao et al. 2020c) designs a representation of three graphs and introduces three aggregators to update message passing for question answering.

Text based image captioning. Image captioning challenges a model to automatically generate a natural language description based on the contents in an image. Existing datasets, *e.g.,* COCO Captions (Chen et al. 2015) and Flickr30k (Young et al. 2014), focus more on visual objects. To enhance text comprehension in the context of an image, a new dataset called TextCaps (Sidorov et al. 2020) is proposed. It requires a model to read and reason about texts and generate coherent descriptions. The baseline model in TextCaps is modified from aforementioned M4C slightly, by removing question input directly.

Generative transformer decoder. To address the problem that answers in these two text-based tasks are usually concatenated by more than one word, we use the structure of transformer (Devlin et al. 2019) decoder in answer module. Following previous work, we also use the generative transformer decoder for fair comparison.

Proposed Method

Given the three modalities (questions, OCR tokens, visual objects), the first step of our model is to prepare the features by projecting them into the same dimension. Then we describe the formulation of *Attention Block* for feature summarizing. Stacking the blocks together yields encoder for downstream tasks. Using encoder output to produce the first word in answer, we then use an iterative decoder to predict the rest of words. This whole process is shown in Figure 3. When transferring to TextCaps, we only make minimal modifications which will be detailed in Section .

Notation In the remainder of this paper, all $\mathbf{W}$ are learned linear transformations, with different symbols to denote independent parameters, *e.g.,* $\mathbf{W}_{fr}$. LN is Layer Normalization (Ba, Kiros, and Hinton 2016). $\circ$ represents element-wise product.

Feature Preparation

Question features. For a question with L words, we first use a three-layer BERT (Devlin et al. 2019) model to embed it into $Q = \{q_i\}_{i=1}^{L}$. This BERT model is finetuned during training.

OCR features. In text-based VQA and image captioning problem, texts are of key importance. Simply gather every feature of text together is not efficient enough. When faced with a bunch of OCRs, human recognition system tends to use two complementary methods to select subsequent words, either by finding similarly-looking and spatially-close words or choosing words coherent in linguistic meaning. For this intuitive purpose, we split features of N OCR tokens into two parts: *Visual* and *Linguistic*.

1) OCR visual-part. Visual features are combined by appearance feature and spatial feature as they show what eyes catch of, without further processing of natural language system. From this part, a model can get visual information such

as word font, color and background. These two are extracted by an off-the-shelf Faster R-CNN (Ren et al. 2015) detector.

$$\mathbf{x}_i^{ocr,v} = \mathrm{LN}(\mathbf{W}_{fr}\mathbf{x}_i^{ocr,fr}) + \mathrm{LN}(\mathbf{W}_{bx}\mathbf{x}_i^{ocr,bx}), \quad (1)$$

where $\mathbf{x}_i^{ocr,fr}$ is the appearance feature extracted from the fc6 layer of Faster R-CNN detector. The fc7 weights are finetuned on our task. $\mathbf{x}_i^{ocr,bx}$ is the bounding box feature in the format of $[x_{tl}, y_{tl}, x_{br}, y_{br}]$, where tl and br denotes top left and bottom right coordinates respectively.

2) OCR linguistic-part. Linguistic features are made up of 1) FastText feature $\mathbf{x}_i^{ocr,ft}$, which is extracted from a pre-trained word embedding and 2) character-level Pyramidal Histogram of Characters (PHOC) (Almazán et al. 2014) feature $x_i^{ocr,ph}$ as they contain natural language related information.

$$\mathbf{x}_i^{ocr,l} = \mathrm{LN}(\mathbf{W}_{ft}\mathbf{x}_i^{ocr,ft} + \mathbf{W}_{ph}\mathbf{x}_i^{ocr,ph}) \quad (2)$$

3) OCR additional features. In the SBD-Trans (Liu et al. 2019; Wang et al. 2019) that we use to recognize OCR tokens, the holistic representations in a specific text region are themselves visual features, however, are employed for linguistic word classification purpose. Therefore, they cover both visual and linguistic context of the OCR token and we thus introduce the Recog-CNN feature $\mathbf{x}_i^{ocr,rg}$ from this network to enrich text features.

Finally, Recog-CNN features are added to OCR visual- and linguistic-part simultaneously.

$$\begin{aligned}\mathbf{x}_i^{ocr,v} &= \mathrm{LN}(\mathbf{W}_{fr}\mathbf{x}_i^{ocr,fr} + \mathbf{W}_{rg}\mathbf{x}_i^{ocr,rg}) + \mathrm{LN}(\mathbf{W}_{bx}\mathbf{x}_i^{ocr,bx}) \\ \mathbf{x}_i^{ocr,l} &= \mathrm{LN}(\mathbf{W}_{ft}\mathbf{x}_i^{ocr,ft} + \mathbf{W}_{ph}\mathbf{x}_i^{ocr,ph} + \mathbf{W}_{rg}\mathbf{x}_i^{ocr,rg})\end{aligned} \quad (3)$$

Visual features. In text-based tasks, visual contents in an image can be utilized to assist textual information in the reasoning process. To prove that our simple attention block has the power of using visual features in various forms, we adopt either grid-based global features or region-based object features.

1) Global features. We obtain image global features x_i^{glob} from a ResNet-152 (He et al. 2016) model pretrained on ImageNet, by average pooling $2048D$ features from the res-5c block, yielding a $14 \times 14 \times 2048$ feature for one image. To be consistent with other features, we resize the feature into 196×2048, a total of 196 uniformly-cut grids.

$$\mathbf{x}_i^{glob} = \mathrm{LN}(\mathbf{W}_g\mathbf{x}_i^{glob}) \quad (4)$$

2) Object features. The region-based object features are extracted from the same Faster R-CNN model as mentioned in OCR features part.

$$\mathbf{x}_i^{obj} = \mathrm{LN}(\mathbf{W}'_{fr}\mathbf{x}_i^{obj,fr}) + \mathrm{LN}(\mathbf{W}'_{bx}\mathbf{x}_i^{obj,bx}), \quad (5)$$

where $\mathbf{x}_i^{obj,fr}$ is the appearance feature and $\mathbf{x}_i^{obj,bx}$ is the bounding box feature.

Attention Block as Feature Summarizing

In tasks that cross the fields of computer vision and natural language processing, modality fusion is of superior importance. Treating them as homogeneous entities in a joint embedding space might be easy to implement, however, is not carefully tailored to a specific problem. Moreover, the many parameters of all entities in the large model (for example, a Transformer) consume much computation. To grasp interaction between modalities for maximum benefit and filter out irrelevant or redundant features before the entering into a large fusion layer, we use a simple attention block to input two sequences of entities and output two processed vectors, which is shown as the *Attention Block* in Figure 3.

The two sequences of entities might be any sequence we want. For TextVQA problem, question changes in real-time and plays a dominant role in final answering. The design of question needs careful consideration and its existence should contribute throughout the process. For example, here we use question as one input of *query* in attention block. The sequence of question words goes through a self-attention process before forwarding into attention block.

First we put the question word sequence $Q = \{q_i\}_{i=1}^{L}$ through a fully connected feed-forward network, which consists of two linear transformations (or two convolutions with 1 as kernel size) and one ReLU activation function between them.

$$q_i^{fc} = \mathrm{conv}\{\mathrm{ReLU}[\mathrm{conv}(q_i)]\}, \ i = 1, \ldots, L; \quad (6)$$

A softmax layer is the used to compute the attention on each word in the question.

$$a_i = \mathrm{Softmax}(q_i^{fc}), \ i = 1, \ldots, L; \quad (7)$$

This is known as self-attention and these weights are multiplied with original question embedding to get weighted sum of word embeddings.

$$Q^s = \textstyle\sum_{i=1}^{L} a_i q_i. \quad (8)$$

If we have several individual entities to combine with question, the corresponding number of parallel self-attention processes are performed on the same question with independent parameters. For example, we can get Q_v^s, Q_l^s and Q_o^s for OCR visual-part, OCR linguistic-part and object regions respectively.

Then the Q^s are used as query for corresponding features. We calculate the attention weights under the guidance of Q^s, which are then put into a softmax layer. Finally the weights are multiplied with original queried features to get a filtered vector. Here we take the pair of Q_v^s and $\mathbf{x}_i^{ocr,v}$ as an example:

$$\begin{aligned} p_i &= \mathbf{W}[\mathrm{ReLU}(\mathbf{W}_s Q_v^s) \circ \mathrm{ReLU}(\mathbf{W}_x x_i^{ocr,v})], \\ s_i &= \mathrm{Softmax}(p_i), \quad i = 1, \ldots, N, \\ g^{ocr,v} &= \textstyle\sum_{i=1}^{N} s_i \mathbf{x}_i^{ocr,v} \end{aligned} \quad (9)$$

where $g^{ocr,v}$ is the output of attention block. Similarly, we can get $g^{ocr,l}$ for the OCR-linguistic summarizing feature, g^{obj} for the object summarizing feature.

Different from M4C sending every single question tokens, OCR tokens and objects into the transformer feature

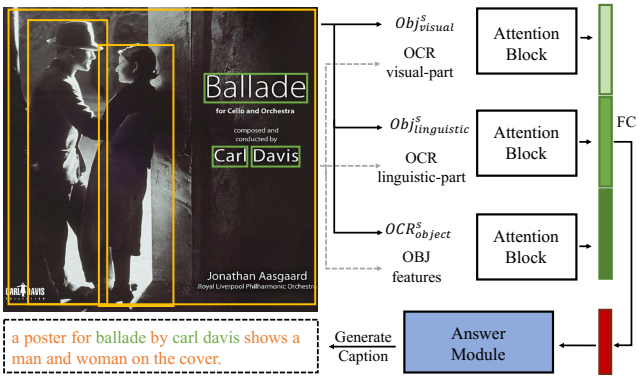

Figure 4: TextCaps baseline model. It has a same structure as TextVQA baseline model.

fusion layer, here we only have 6 feature vectors (Q^s_v, Q^s_l, Q^s_o, $g^{ocr,v}$, $g^{ocr,l}$ and g^{obj}) which are sent to the following process. This largely decreases the computation complexity and burden, considering that the transformer is a parameter-heavy network.

Stacked-Block Encoder

The attention block in Section can be stacked together as an encoder which produces combined embedding for downstream tasks.

TextVQA baseline model. As presented in the above module, questions are sent through self-attention to output Q^s_v, Q^s_l and Q^s_o. OCR features in images are splitted into visual- and linguistic part, which are $x^{ocr,v}$ and $x^{ocr,l}$. We also have object features x^{obj}. The six sequences are put into three attention blocks and we get six $768D$ vectors which are then forwarded into a fusion encoder. The fusion encoder, OCR encoder and generative decoder in Figure 3 are in the same transformer model but undertaking different roles. After fusion encoder processing, the six vectors conduct element-wise multiplication in a pairwise way to get corresponding embeddings which are concatenated together. Then we use a fully-connected layer to transform the concatenated embeddings to a context embedding with appropriate dimension, upon which we generate the first answer output. Given the first answer word, a generative decoder is then used to select the rest of answer, which will be detailed in Section and Supplementary Material.

TextCaps baseline model. As there are no questions in TextCaps, we use objects to guide OCR visual- and linguistic-part and use OCRs to guide object features. Technically we simply replace question word sequence with OCR token sequence or object proposal feature sequence. The other settings are the same with TextVQA. Figure 4 illustrates our Textcaps baseline model. To easily transfer to another task demonstrates the generalization ability and simplicity of our method.

Answer Generation Module

To answer a question or generate a caption, we use a generative decoder based on transformer. It takes as input the 'context embedding' from the above encoder and select the first word of the answer. Based on the first output word, we then use the decoder to find the next word token either from a pre-built vocabulary or the candidate OCR tokens extracted from the given image, based on a scoring function.

Training Loss. Considering that the answer may come from two sources, we use multi-label binary cross-entropy (bce) loss:

$$\begin{aligned} pred &= \frac{1}{1+\exp\left(-y_{pred}\right)}, \\ \mathcal{L}_{bce} &= -y_{gt}\log(pred) - (1-y_{gt})\log(1-pred), \end{aligned} \tag{10}$$

where y_{pred} is prediction and y_{gt} is ground-truth target.

Additional Training Loss. In some cases, the model reasons correctly, however, picks slightly different words than what we expected due to defective reading (OCR) ability. To take advantage of these predictions, we introduce a new policy gradient loss as an auxilliary task inspired by reinforcement learning. In this task, the greater reward, the better. We take Average Normalized Levenshtein Similarity (ANLS) metric[1] as the reward which measures the character similarity between predicted answer and ground-truth answer.

$$\begin{aligned} r &= \text{ANLS}(\phi(y_{gt}), \phi(y_{pred})), \\ y &= 1(\text{softmax}(y_{pred})), \\ \mathcal{L}_{pg} &= (0.5-r)(y_{gt}\log(y) + (1-y_{gt})\log(1-y)), \\ \mathcal{L} &= \mathcal{L}_{bce} + \alpha \cdot \mathcal{L}_{pg}, \end{aligned} \tag{11}$$

where ϕ is a mapping function that returns a sentence given predicting score (*e.g.*, , y_{pred}), ANLS$(\cdot)$ is used to calculate similarity between two phrases, 1 is an indicator function to choose the maximum probability element. The additional training loss is a weighted sum of $\mathcal{L}_{bce}$ and $\mathcal{L}_{pg}$, where α is a hyper-parameter to control the trade-off of $\mathcal{L}_{pg}$. After introducing policy gradient loss, our model is able to learn fine-grained character composition alongside linguistic information. We only apply this additional loss on ST-VQA dataset, which brings roughly 1% improvement.

Experiments

Extensive experiments are conducted across two categories of tasks: TextVQA and TextCap. For TextVQA we set new state-of-the-art on TextVQA dataset and all three tasks of ST-VQA. For TextCaps we surpass 2020 TextCaps Challenge winner. See more experiments details below.

Implementation Details

The set of methods are built on top of PyTorch. We use Adam as the optimizer. The learning rate for TextVQA and TexCaps is set to $1e{-}4$. For TextVQA we multiply the learning rate with a factor of 0.1 at the $14,000$ and $15,000$ iterations in a total of $24,000$ iterations. For TextCaps the multiplication is done at the $3,000$ and $4,000$ iterations, with

[1] $\text{ANLS}(s_1, s_2) = 1 - d(s_1, s_2)/\max(\text{len}(s_1), \text{len}(s_2))$, d(,) is edit distance.

#	Method	Trans structure	OCR system	Visual feat	OCR feat	Accu. on val	Accu. on test
1	one-block	4-layer	Rosetta-en	-	Fast, PHOC, FRCN, bbox	37.51	-
2	two-block	4-layer	Rosetta-en	-	Fast, PHOC, FRCN, bbox	39.28	39.99
3	three-block	4-layer	Rosetta-en	Global	Fast, PHOC, FRCN, bbox	39.52	-
4	three-block	4-layer	Rosetta-en	Obj	Fast, PHOC, FRCN, bbox	39.91	-
5	three-block	4-layer	Rosetta-en	Obj	Fast, PHOC, FRCN, bbox, Recog-CNN	40.28	-
6	three-block	8-layer	Rosetta-en	Obj	Fast, PHOC, FRCN, bbox, Recog-CNN	40.38	40.92
7	three-block	8-layer	SBD-Trans	Obj	Fast, PHOC, FRCN, bbox, Recog-CNN	43.95	44.72
8	three-block(w/ST-VQA)	8-layer	SBD-Trans	Obj	Fast, PHOC, FRCN, bbox, Recog-CNN	**45.53**	**45.66**

Table 1: We ablate our model on TextVQA dataset by testing number of attention blocks, forms of visual object features and addition of OCR representations.

#	Method	OCR system	Accu. on val	Accu. on test
1	LoRRA (Singh et al. 2019)	Rosetta-ml	26.56	27.63
2	DCD ZJU (Lin et al. 2019)	-	31.48	31.44
3	MSFT VTI	-	32.92	32.46
4	M4C (Hu et al. 2019)	Rosetta-en	39.40	39.01
5	SA-M4C (Kant et al. 2020)	Google OCR	45.40	44.60
6	SMA (Gao et al. 2020a)	SBD-Trans	44.58	45.51
7	ours (three-block)	Rosetta-en	40.38	40.92
8	ours (three-block w/ST-VQA)	SBD-Trans	**45.53**	**45.66**

Table 2: Comparison to previous work on TextVQA dataset. Our model sets new state-of-the-art with an extremely simple design.

#	Method	ANLS on test1	ANLS on test2	ANLS on test3
1	M4C (Hu et al. 2019)	-	-	0.4621
2	SA-M4C (Kant et al. 2020)	-	0.4972	0.5042
3	SMA (Gao et al. 2020a)	0.5081	0.3104	0.4659
4	ours	0.5060	0.5047	0.5089
5	ours(w/TextVQA)	**0.5490**	**0.5513**	**0.5500**

Table 3: Comparison to previous work on ST-VQA dataset. With TextVQA pretraining, our model outperforms current approaches by a large margin.

$12,000$ total iterations. We set the maximum length of questions to $L = 20$. We recognize at most $N = 50$ OCR tokens and detect at most $M = 100$ objects. The maximum number of decoding steps is set to 12. Transformer layer in our model uses 12 attention heads. The other hyper-parameters are the same with BERT-BASE (Devlin et al. 2019). We use the same model on TextVQA and three tasks of ST-VQA, only with different answer vocabulary, both with a fixed size of 5000.

Ablation Study on TextVQA Dataset

TextVQA (Singh et al. 2019) is a popular benchmark dataset to test scene text understanding and reasoning ability, which contains $45,336$ questions on $28,408$ images. Diverse questions involving inquires about time, names, brands, authors, *etc.,* and dynamic OCR tokens that might be rotated, casual or partially occluded make it a challenging task.

We first conduct an experiment of building only one block, with one question self-attention output guiding the whole set of text features as a comparison. This is a **one-block** model in Table1 which does not perform as good as the state-of-the-art. To investigate how well text features can perform without the usage of visual grid-based or region-based features, we build a **two-block** model. Given the two categories of OCR features – visual and linguistic, we find that our simple model is already able to perform promisingly on TextVQA problem. Line 1 and Line 2 tell clearly the validity of sorting text features into two groups (1.77% difference).

When building our third block on the basis of visual contents, either global features or object-level features are at our disposal. The incorporation of a third block has modest improvements (0.24% for global and 0.63% for object features). From Line 4 to Line 5, a new Recog-CNN feature is added to enrich text representation and brings 0.37% improvement. We also use more transformer layers (from 4 to 8) and get 0.1% higher result. Then we use a much better OCR system (especially on recognition part) and obtain large performance boost (from 40.38% to 43.95%).

Qualitative examples. We present four examples in Figure 5 where our model shows the ability to really read scene texts in a way similar to humans, *i.e.,* left-to-right then top-to-bottom. In contrast, current state-of-the-art model M4C fails to read tokens in a correct order.

Comparison with State-of-the-art

TextVQA dataset. Even stripped of the usage of visual contents in the image, our two-block model already surpasses current state-of-the-art M4C by 0.98% on test set (Line 2 in Table 1 VS. Line 4 in Table 2). Using the same OCR system, our baseline model further improves upon M4C by 0.98% on val and 1.91% on test (Line 7 VS. Line 4 in Table 2).

Compared to the top entries in TextVQA Challenge 2020, our baseline has a significantly simpler model design, especially on the encoder side. M4C and SA-M4C take all parameters of entities into transformer layers and join large amout of computation. SMA uses a heterogeneous graph

Q: what does coca cola do?

A(ours): relieves fatigue
A(M4C): southern home

Q: what side is this sign pointing to?

A(ours): west side
A(M4C): side

Q: who makes this product?

A(ours): airport extreme
A(M4C): airport wi-fi

Q: what is the name of the business?

A(ours): airport business centre
A(M4C): airport centre

Figure 5: Qualitative examples of our baseline model in contrast to M4C. While our model can read texts in accordance with written language system, M4C can only pick tokens in a random and errorneous way.

#	Method	Val set metrics				
		B	M	R	S	C
1	M4C-Captioner	23.30	22.00	46.20	15.60	89.60
2	ours(Rosetta-en)	23.87	22.02	46.4	15.01	91.06
3	ours(SBD-Trans)	**24.89**	**22.71**	**47.24**	**15.71**	**98.83**
#	Method	Test set metrics				
		B	M	R	S	C
4	M4C-Captioner	18.9	19.8	43.2	12.8	81.0
5	colab_buaa(Winner)	20.09	**20.62**	**44.30**	**13.50**	88.48
6	ours(SBD-Trans)	**20.16**	20.3	44.23	12.82	**89.63**

Table 4: Results on TextCaps dataset. (B: BLEU-4; M: METEOR; R: ROUGE_L; S: SPICE; C: CIDEr)

to explicitly consider different nodes and compute attention weights on 5-neighbored graph. Our model that surpasses all of them only sends six holistic vectors two-by-two into transformer layers, which tremendously saves computation.

ST-VQA dataset. The ST-VQA dataset (Biten et al. 2019) is another popular dataset with three tasks, which gradually increase in difficulty. Task 1 provides a dynamic candidate dictionary of 100 words per image, while Task 2 provides a fixed answer dictionary of $30,000$ words for the whole dataset. As for Task 3, however, the model are supposed to generate answer without extra information. We also evaluate our model on ST-VQA dataset, using the same model from TextVQA for all three tasks. Without any additional training data, our model achieved the highest on Task 2 and Task 3 (Line 4 in Table 3). Using TextVQA dataset as additional training data, our model sets new state-of-the-art on all three tasks and outperforms current approaches by a large margin.

TextCaps Dataset

TextCaps is a new dataset that requires a model to read texts in images and generate descriptions based on scene text understanding and reasoning. In TextCaps, automatic captioning metrics (BLEU (Papineni et al. 2002), METEOR (Denkowski and Lavie 2014), ROUGE_L (Lin 2004), SPICE (Anderson et al. 2016) and CIDEr (Vedantam, Lawrence Zitnick, and Parikh 2015)) are compared with human evaluation scores. All automatic metrics show high correlation with human scores, among which CIDEr and METEOR have the highest.

M4C-Captioner is the method provided in TextCaps, which is modified from M4C model by simply removing question input. Similarly, simply replacing question in our TextVQA baseline model with object or OCR sequence yields our TextCaps baseline model. Using exactly the same OCR system, OCR representations, our baseline with Rosetta-en OCR (Line 2 on Table 4) already surpass M4C-Captioner (Line 1 on Table 4), especially on BLUE-4 and CIDEr metric. By upgrading our OCR system to SBD-Trans and using 6-layer transformer in our encoder-decoder structure, our baseline further exceeds TextCaps Challenge Winner on BLUE-4 and CIDEr metric as shown in Line 5 and Line 6 of Table 4.

Conclusion

In this paper, we provide a simple but rather strong baseline for the text-based vision-and-language research. Instead of handling all modalities over a joint embedding space or via complicated graph structural encoding, we use the vanilla attention mechanism to fuse pairwise modalities. We further split text features into two functionally different parts, *i.e.,* linguistic- and visual-part which flow into corresponding attention branch. We evaluate our simple baseline model on TextVQA, ST-VQA and TextCaps, all leading to the best performance on the public leaderboards. This sets the new state-of-the-art and our model could be the new backbone model for both TextVQA and TextCaps. What's more, we believe this work inspires a new thinking of the multi-modality encoder design.

Acknowledgements

This work was supported by the Ministry of Science and Technology of China (No. 2020AAA0106900), and the National Natural Science Foundation of China (No.61876152, No.U19B2037).

References

Almazán, J.; Gordo, A.; Fornés, A.; and Valveny, E. 2014. Word spotting and recognition with embedded attributes. *IEEE transactions on pattern analysis and machine intelligence* 36(12): 2552–2566.

Anderson, P.; Fernando, B.; Johnson, M.; and Gould, S. 2016. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision*, 382–398. Springer.

Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang., L. 2018. Bottom-up and top-down attention for image captioning andvisual question answering. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Lawrence Zitnick, C.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2425–2433.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* .

Biten, A. F.; Tito, R.; Mafla, A.; Gomez, L.; Rusinol, M.; Valveny, E.; Jawahar, C.; and Karatzas, D. 2019. Scene Text Visual Question Answering. In *Proc. IEEE Int. Conf. Comp. Vis.*

Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325* .

Denkowski, M.; and Lavie, A. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, 376–380.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. 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* .

Gao, C.; Zhu, Q.; Wang, P.; Li, H.; Liu, Y.; Hengel, A. v. d.; and Wu, Q. 2020a. SMA_NWPU_Adelaide_Team , TextVQA Challenge 2020 winner. https://visualqa.org/workshop.html.

Gao, C.; Zhu, Q.; Wang, P.; Li, H.; Liu, Y.; Hengel, A. v. d.; and Wu, Q. 2020b. Structured Multimodal Attentions for TextVQA. *arXiv preprint arXiv:2006.00753* .

Gao, D.; Li, K.; Wang, R.; Shan, S.; and Chen, X. 2020c. Multi-Modal Graph Neural Network for Joint Reasoning on Vision and Scene Text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

Hu, R.; Singh, A.; Darrell, T.; and Rohrbach, M. 2019. Iterative Answer Prediction with Pointer-Augmented Multimodal Transformers for TextVQA. *arXiv preprint arXiv:1911.06258* .

Johnson, J.; Hariharan, B.; van der Maaten, L.; Fei-Fei, L.; Lawrence Zitnick, C.; and Girshick, R. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2901–2910.

Kahou, S. E.; Michalskiand, V.; Atkinson, A.; Kadar, A.; Trischler, A.; and Bengio, Y. 2018. FigureQA: An annotated figure dataset for visual reasoning. In *ICLR workshop track*.

Kant, Y.; Batra, D.; Anderson, P.; Schwing, A.; Parikh, D.; Lu, J.; and Agrawal, H. 2020. Spatially Aware Multimodal Transformers for TextVQA. In *Proc. Eur. Conf. Comp. Vis.*

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Lin, Y.; Zhao, H.; Li, Y.; and Wang., D. 2019. DCD_ZJU, TextVQA Challenge 2019 winner. https://visualqa.org/workshop.html.

Liu, Y.; Zhang, S.; Jin, L.; Xie, L.; Wu, Y.; and Wang, Z. 2019. Omnidirectional Scene Text Detection with Sequential-free Box Discretization. In *Proc. Int. Joint Conf. Artificial Intell.*, 3052–3058.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, 91–99.

Sidorov, O.; Hu, R.; Rohrbach, M.; and Singh, A. 2020. TextCaps: a Dataset for Image Captioning with Reading Comprehension. In *Proc. Eur. Conf. Comp. Vis.*

Singh, A.; Natarajan, V.; Shah, M.; Jiang, Y.; Chen, X.; Batra, D.; Parikh, D.; and Rohrbach, M. 2019. Towards VQA Models That Can Read. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*

Vedantam, R.; Lawrence Zitnick, C.; and Parikh, D. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4566–4575.

Wang, P.; Wu, Q.; Shen, C.; Dick, A.; and van den Hengel, A. 2018. Fvqa: Fact-based visual question answering. *IEEE Trans. Pattern Anal. Mach. Intell.* 40(10): 2413–2427.

Wang, P.; Yang, L.; Li, H.; Deng, Y.; Shen, C.; and Zhang, Y. 2019. A Simple and Robust Convolutional-Attention Network for Irregular Text Recognition. *arXiv:1904.01375* .

Young, P.; Lai, A.; Hodosh, M.; and Hockenmaier, J. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics* 2: 67–78. doi:10.1162/tacl_a_00166. URL https://www.aclweb.org/anthology/Q14-1006.