

Question-controlled Text-aware Image Captioning

Anwen Hu
School of Information
Renmin University of China
anwenhu@ruc.edu.cn

Shizhe Chen
INRIA
shizhe.chen@inria.fr

Qin Jin*
School of Information
Renmin University of China
qjin@ruc.edu.cn

ABSTRACT

For an image with multiple scene texts, different people may be interested in different text information. Current text-aware image captioning models are not able to generate distinctive captions according to various information needs. To explore how to generate personalized text-aware captions, we define a new challenging task, namely Question-controlled Text-aware Image Captioning (Qc-TextCap). With questions as control signals, this task requires models to understand questions, find related scene texts and describe them together with objects fluently in human language. Based on two existing text-aware captioning datasets, we automatically construct two datasets, ControlTextCaps and ControlVizWiz to support the task. We propose a novel Geometry and Question Aware Model (GQAM). GQAM first applies a Geometry-informed Visual Encoder to fuse region-level object features and region-level scene text features with considering spatial relationships. Then, we design a Question-guided Encoder to select the most relevant visual features for each question. Finally, GQAM generates a personalized text-aware caption with a Multimodal Decoder. Our model achieves better captioning performance and question answering ability than carefully designed baselines on both two datasets. With questions as control signals, our model generates more informative and diverse captions than the state-of-the-art text-aware captioning model. Our code and datasets are publicly available at <https://github.com/HAWLYQ/Qc-TextCap>.

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation; Computer vision.**

KEYWORDS

Image Captioning; Scene Text; Question-controlled

1 INTRODUCTION

Texts are omnipresent and convey valuable information of the visual environment, such as the title of a book, the time shown on a clock and the words on a road sign. With the goal of describing the visual world to visually impaired people, it is essential to comprehend such scene texts beyond pure visual recognition [2, 10, 14, 18, 23, 28, 32]. Therefore, more recent works are focusing on the text-aware image captioning task [24, 29, 31, 33, 35], which aims to describe an image in natural sentences covering scene text information in the image.

However, when an image consists of rich scene text information as shown in Figure 1, it can be tedious and often not necessary to describe all of the texts in the image. According to a user study in [19], visually impaired people prefer to know their surrounding environment i.e. images in a progressive manner. For example in

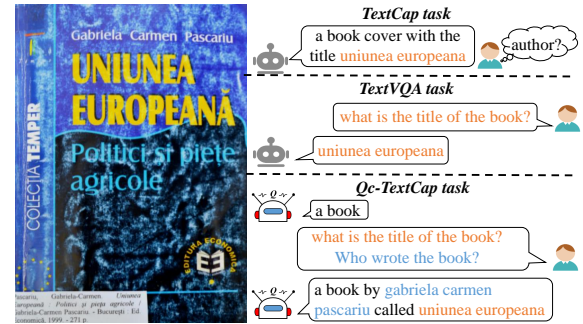


Figure 1: An example of Question-controlled Text-aware Image Captioning (Qc-TextCap). The corresponding question and answer are labelled with the same color.

Figure 1, an image captioning system is more preferable to firstly tell the visually impaired users an overview description of the image i.e. “a book”, and then let the user interact with the system to obtain more specific details about their interested scene texts, such as “who wrote the book” or “what is the title of the book”. In this way, the visually impaired users can obtain more personalized text-aware captions according to their interests.

To satisfy such user needs, in this work, we propose a novel task called Question-controlled Text-aware Image Captioning (Qc-TextCap), where users can ask scene text related questions based on an initial coarse-grained image description to obtain a more informative text-aware image description for the questions. Compared to the Text Visual Question Answering (TextVQA) task [4, 13, 15, 25] which only contains a single question and a single word answer without initial caption, our Qc-TextCap task demands higher multimodal comprehension abilities. For example, the user can simultaneously ask multiple questions and the model should organize all scene text answers associated with their corresponding visual objects in a fluent sentence description as shown in Figure 1. Compared to previous endeavors in controllable image captioning such as providing object tokens [34], image regions [6], and abstract scene graphs [5] to generate different captions, using questions as the control signal is a more convenient and natural interaction interface for the visually impaired people. Whereas previous control signals require users to specify which visual objects or regions he/she wants to know.

As existing image caption datasets and VQA datasets do not directly support the Qc-TextCap task, we propose an automatic approach to construct data in the form of quadruples, $\langle \text{image, initial simple caption, control questions, text-aware captions} \rangle$, from existing text-aware caption datasets i.e. TextCaps [24] and VizWiz-Captions [11]. Specifically, given annotated text-aware captions

*Corresponding Author

for images in existing datasets, we use automatic approaches to generate an initial caption and questions related to scene texts in the image. Based on the Qc-TextCap dataset, we further propose a Geometry and Question Aware Model (GQAM) for the Question-controlled Image Captioning task. The GQAM consists of three key modules, namely a Geometry-informed Visual Encoder that fuses object region features with scene text region features considering their spatial relationship; a Question-guided Encoder that attends to corresponding relevant visual features to encode word-level question information; and a Multimodal Decoder that takes the visual region features, question features and initial caption features as input to generate a text-aware caption. Experimental results on two datasets demonstrate that our model is able to effectively generate text-aware captions to answer different questions.

The main contributions of our work are as follows:

- We propose a novel challenging task, namely Question-controlled Text-aware Image Captioning (Qc-TextCap), towards generating informative and personalized image captions to benefit visually-impaired users.
- We develop an automatic system to construct two appropriate datasets for the Qc-TextCap task based on existing text-aware image captioning datasets.
- We propose a novel captioning model GQAM that progressively encodes relevant multimodal features with Geometry-informed Visual Encoder and Question-guided Encoder and generates informative captions via a Multimodal Decoder.
- GQAM outperforms carefully designed baselines in Qc-TextCap and generates more informative and diverse captions than the text-aware captioning model.

2 RELATED WORK

General Image Captioning. Lots of neural network based models [2, 10, 12, 14, 18, 23, 28, 32] have been proposed for general image captioning. AoANet [14] achieves state-of-the-art performance with an attention on attention mechanism. Fishch *et al.* [10] propose to use question answering accuracy as reward during training to increase the amount of information in generated captions. During inference, their method still simply generates a single caption without questions as control signals. These methods are able to enumerate major objects and describe their relationships but fail to provide detailed text information in the image.

Text-aware Image Captioning. Text-aware Image Captioning aims to comprehend text information in an image and relate it to visual objects. TextCaps [24] and VizWiz-Captions [11] are two available datasets. Images of TextCaps come from Open Images V3 dataset and are all verified to contain scene texts. Images in VizWiz-Captions are taken by the blind and around 63% images include scene texts. Images taken by the blind may have quality issues, such as overexposure, but can better represent the real use case for visually-impaired people. Based on these two datasets, there have been some models [29, 31, 33, 35] proposed to improve the quality of text-aware captions. Wang *et al.* [29] propose to encode intrinsic spatial relationship between OCR tokens to generate more complete scene text information. Zhu *et al.* [35] propose a simple strong baseline which consists of multiple attention blocks. Wang *et al.* [31] introduce confidence embedding of OCR tokens to help

select the most noteworthy scene texts. Yang *et al.* [33] design text-aware pre-training tasks to enhance the model ability in reading and understanding scene texts. These works contribute to generating a better text-aware caption for each image but have not explored how to generate personalized captions.

Text Visual Question Answering. Models for TextVQA [4, 13, 15, 25] are designed to find correct scene texts from images to answer questions. There are two major differences between TextVQA task and Qc-TextCap task. First, Qc-TextCap requires models to process multiple questions simultaneously. Second, models for Qc-TextCap should be able to organize multiple scene text answers and relevant objects for a fluent description.

Controllable Image Captioning. Controllable Image Captioning task aims to generate captions in different styles or capturing different contents in the image [5, 6, 34]. Zheng *et al.* [34] propose to describe an image with a guiding object as the initial token. Cornia *et al.* [6] use image regions as control signals to produce multiple captions for a given image. Chen *et al.* [5] further design abstract scene graphs to represent user intention in fine-grained level. Concrete control signals used in these works can clearly guide model which part of an image to describe. However, only when users are able to see the image can they give such kinds of signals, which is unrealistic for the blind. In this paper, we first propose to use human language as the control signal, which is abstract but more suitable for visually-impaired people to interact with the machine.

3 QC-TEXTCAP DATASET

The Question-controlled Text-aware Image Captioning (Qc-TextCap) task simulates realistic scenario to generate personalized captions for visually impaired people. An automatic image captioning system firstly generates a general caption C^{ini} about an image I for the user, which describes major objects in the image but contains no scene text information. Then the user can ask scene text related questions Q to obtain more specific information. The system aims to re-generate a new text-aware caption Y to answer questions Q . However, existing text-aware image captioning datasets (i.e. TextCaps [24] and VizWiz-Captions [11]) only contain $\langle I, Y \rangle$ pairs, which do not support such interactive control of caption generation. Therefore, in this section, we present an automatic approach to build $\langle I, C^{ini}, Q, Y \rangle$ data samples based on available $\langle I, Y \rangle$ annotations for the Qc-TextCap task.

3.1 Automatic Dataset Construction

Figure 2 illustrates the pipeline of our automatic dataset construction approach, consisting of *Initial Caption Generation* and *Question Generation* steps. The *Initial Caption Generation* step generates initial general captions about the image, which only contain visual object information without any scene texts. The *Question Generation* step is to produce questions about scene texts in the image.

1) Initial Caption Generation. The initial captions should only express essential image contents such as major visual objects without tedious details, especially scene texts in the image. Though it is possible to generate initial captions via directly applying an image captioning model pre-trained on general captioning datasets such as MSCOCO [17], it is not optimal for our task due to domain differences. Images in MSCOCO datasets are web images carefully taken

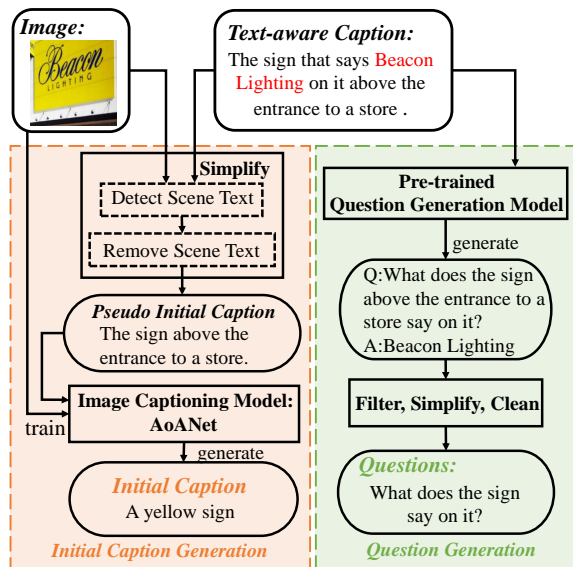


Figure 2: The process of automatic dataset construction for Question-controlled Text-aware Captioning.

by photographers, while we focus on images containing scene texts which may even be taken by the blind. Besides, the initial captions would be better to mention scene text relevant objects for the task. Hence, we first extract in-domain pseudo initial captions \tilde{C}^{ini} based on I and Y , and use them to train an image captioning model to generate initial caption C^{ini} automatically.

Specifically, we apply two steps to obtain \tilde{C}^{ini} : detecting scene texts in Y and then removing the detected scene texts. Firstly, we detect which words in the text-aware caption Y are scene text words. We perform Optical Character Recognition (OCR) on images via Microsoft Azure Read API¹ and compare the OCR results with words in the caption. As automatic OCR results are not perfect especially for low-quality images taken by the blind, we further perform a series of Natural Language Processing (NLP) procedures to improve the scene text detection in caption. If a phrase is recognized as a Named Entity (e.g. a person name or a company name), it is considered as a scene text even though it does not match any OCR results. After detecting scene texts in Y , we need to remove them without hurting the fluency of the sentence. Naively deleting the words is prone to make grammatical errors. Therefore, we use the syntactic dependency parser Spacy² to construct a dependency tree and prune the branches containing the scene texts. We present an example to illustrate this process in the supplementary material. In this way, we can obtain pseudo initial captions \tilde{C}^{ini} .

Then we train an automatic image captioning model given I and \tilde{C}^{ini} pairs on the training set. Specifically, we use the state-of-the-art AoANet [14] model. The in-domain training enables AoANet model to generate C^{ini} that mentions major objects in the image. Besides, the similarity between C^{ini} and Y is much lower than \tilde{C}^{ini}

¹<https://docs.microsoft.com/en-us/azure/cognitive-services/computer-vision/overview-ocr>

²<https://spacy.io/>

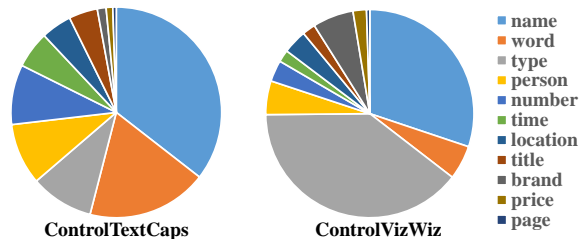


Figure 3: Question type distribution in the ControlTextCaps and ControlVizWiz datasets.

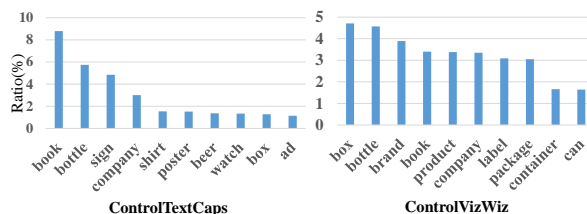


Figure 4: Top 10 objects in the questions of ‘name’ type.

which is better to simulate real applications where the initial caption may not be perfect. More details about the training of AoANet are provided in our supplementary material.

2) Question Generation. The goal is to generate scene text related questions as control signals. We first use T5 model [21] trained on SQuADv1 [22] dataset to generate multiple questions and answers given the target caption Y . As some QA pairs are not relevant to scene texts, we filter out such QA pairs by checking whether the answers can be found in OCR results. Besides, questions about one scene text may contain scene text answers of other questions or extra object descriptions not in initial captions, which leak groundtruth information to the model and should be avoided. Therefore, we further apply several question cleaning steps to remove such extra leaking information. As shown in Figure 2, for the initial caption ‘a yellow sign’, a question ‘what does the sign above the entrance to a store say on it’ is transferred into ‘what does the sign say on it’. More details about question filtering and question cleaning can be found in the supplementary material.

3.2 Dataset Analysis

We automatically construct two Qc-TextCap datasets based on TextCaps [24] and VizWiz-Caption [11], namely ControlTextCaps and ControlVizWiz. Each example is a tuple of an image (I), a target text-aware caption (Y), an automatic initial caption (C^{ini}), a pseudo initial caption (\hat{C}^{ini}), and several questions (Q) towards scene texts in Y . The statistics of these two datasets are presented in Table 1. Each image in raw datasets is annotated with multiple text-aware captions, so an image may appear in multiple tuples in our Qc-TextCap datasets.

Initial Caption Quality. In terms of sequence length, the text-aware caption Y is around 71% longer than \tilde{C}^{ini} and 120% longer than C^{ini} . To better quantify the quality of the pseudo initial caption

Table 1: Datasets statistics of our ControlTextCaps and ControlVizWiz. $Tuple = (I, Y, \tilde{C}^{ini}, C^{ini}, Q)$, where $I, Y, \tilde{C}^{ini}, C^{ini}$ represent image, target caption, pseudo initial caption and automatic initial caption respectively, $Q = \{Q\}$ represents questions related to the target caption Y , O refers to OCR tokens. $N(x)$ is the number of x . $L(x)$ is the average sequence length of x . $p^{obj}(C^{ini})$ means the precision of objects in automatic initial captions.

Dataset	Split	$N(Tuple)$	$N(Q)$	$N(I)$	$L(Y)$	$L(Q)$	$L(O)$	$L(\tilde{C}^{ini})$	$L(C^{ini})$	$CIDEr(\tilde{C}^{ini})$	$CIDEr(C^{ini})$	$p^{obj}(C^{ini})$
ControlTextCaps	train	65,192	73,719	20,217	12.3	7.7	13.5	7.2	5.6	385.73	52.99	94.90
	validation	4,992	5,468	1,571	12.0	7.9	13.5	7.4	5.9	437.92	38.92	79.90
	test	5,000	5,753	1,429	11.9	7.5	15.1	6.8	5.3	386.11	32.73	82.04
ControlVizWiz	train	25,339	29,139	10,763	11.8	7.4	11.3	7.0	5.2	379.32	44.34	85.25
	validation	1,962	2,252	805	11.7	7.4	11.6	6.8	5.2	387.14	32.54	70.31
	test	1,956	2,258	839	11.8	7.5	12.2	7.1	5.4	406.28	32.68	66.51

\tilde{C}^{ini} and automatic initial caption C^{ini} , we calculate CIDEr [27] scores of them against the ground truth text-aware captions Y . As shown in Table 1, \tilde{C}^{ini} achieves very high CIDEr scores, which indicates that pseudo initial captions are very similar with target captions. Using them as initial captions might not be good for developing the model ability in enriching object description. The automatic initial caption has much lower CIDEr scores compared to the pseudo initial caption, but achieves acceptable performance in object precision. Therefore, the automatic initial caption is more suitable to be used as the initial caption that describes major objects but gives little hint about target text-aware caption.

Question Quality. We ask humans to evaluate the quality of automatically generated questions. For each question, we ask a human evaluator to classify it as ‘No error’, ‘Grammar error’ or ‘Semantic error’. ‘Grammar error’ means the question is indeed asked for scene texts in the corresponding caption but contains some grammar errors. ‘Semantic error’ means the question is irrelevant with the scene text. We ask 10 people to evaluate 2070 questions in ControlTextCaps and 4 people to evaluate 850 questions in ControlVizWiz. According to the statistics, the ‘Semantic error’ only accounts for 13.48% and 20.35% in ControlTextCaps and ControlVizWiz, respectively. ‘No error’ accounts for 62.90% and 55.53%. This indicates that these automatically generated questions are good enough to support Qc-TextCap task.

Question Diversity. Question diversity is critical to simulate different kinds of information needs in real use case. To measure the question diversity of these two datasets, we extract the backbone of questions by Syntactic Dependency Parsing and then classify them to 11 question types by rules. Figure 3 presents the question type distribution on two Qc-TextCap datasets. Specifically, questions of ‘name’ type account more than 30% in both two datasets. To analyze this type of question in a fine-grained way, we further distinguish them according to the queried objects. Figure 4 shows ratio of top 10 queried objects in ‘name’ questions. First, in both datasets, questions about top objects account for a small proportion, which indicates that the objects queried in ‘name’ questions are diverse. Second, top objects queried in ControlTextCaps and ControlVizWiz are different. Questions in ControlVizWiz are more about the objects held by hand, which is consistent with the fact that pictures in VizWiz are all taken by blind people.

4 QC-TEXTCAP MODEL

In this section, we introduce the Geometry and Question Aware Model (GQAM) for the Qc-TextCap task. As illustrated in Figure 5, GQAM consists of three modules, namely Geometry-informed Visual Encoder, Question-guided Encoder and Multimodal Decoder. The Geometry-informed Visual Encoder fuses object region features and scene text region features with relative geometry information. Question-guided Encoder dynamically selects relevant visual features to encode questions. The Multimodal Decoder takes inputs of visual, question and initial caption features to sequentially generate a text-aware caption for the question.

Given input I, C^{ini}, Q , we use bottom-up-attention [2] model to detect object bounding boxes $B^{obj} = [b_1^{obj}, \dots, b_{N_{obj}}^{obj}]$ and Microsoft Azure Read API to detect scene text bounding boxes $B^{ocr} = [b_1^{ocr}, \dots, b_{N_{ocr}}^{ocr}]$. Then we extract both object region features $V^{obj} = [v_1^{obj}, \dots, v_{N_{obj}}^{obj}]$ and scene text region features $V^{ocr} = [v_1^{ocr}, \dots, v_{N_{ocr}}^{ocr}]$ by bottom-up-attention model. For the initial caption C^{ini} and questions Q , we extract token-level embeddings $T^{ini} = [t_1^{ini}, t_2^{ini}, \dots, t_{N_{ini}}^{ini}]$ and $T^{que} = [t_1^{que}, t_2^{que}, \dots, t_{N_{que}}^{que}]$ (multiple questions in Q are concatenated to one token sequence) with a trainable three-layer transformer, which is initialized from the first 3 layers of Bert_{base} [9].

4.1 Geometry-informed Visual Encoder

Spatial relation between object regions and scene text regions is critical for accurately describing the scene text information about an object. For example, for a question ‘what is the title of the book’, the region containing ‘title’ information is certainly included in the region of the ‘book’ object. Besides, encoding the relative position between object regions is also beneficial for describing the object relationships in captions. Furthermore, due to typesetting, long phrases in images are usually separated into multiple rows or columns. By utilizing the spatial relation between scene text regions, visual features of one phrase can be encoded together. Therefore, we apply a module to jointly encode object region features and scene text region features $V = [V^{obj}, V^{ocr}]$ with the help of geometry information $B = [B^{obj}, B^{ocr}]$, namely Geometry Informed Visual Encoder.

The visual encoder is based on multi-head attention [26]. We propose a geometry self-attention mechanism to influence the visual self-attention distribution among both object regions and scene text regions. The geometry informed self-attention is calculated as

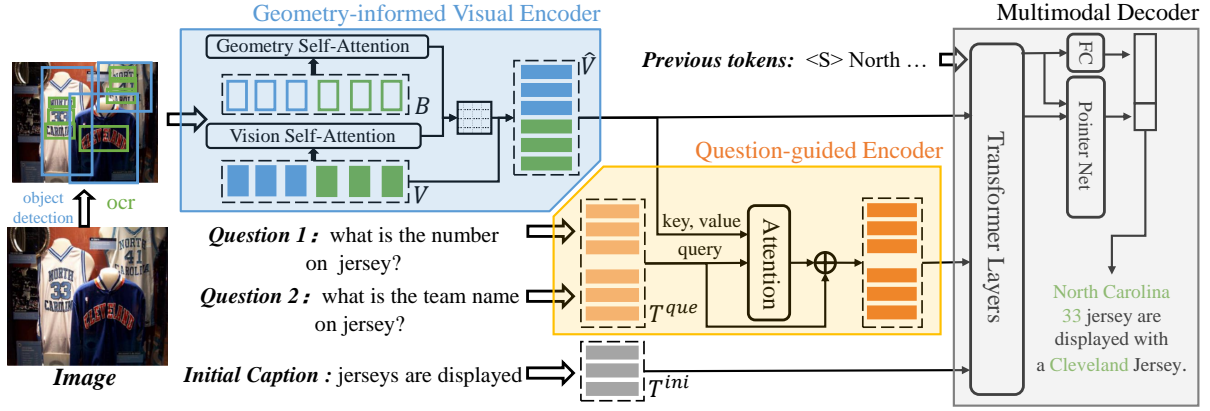


Figure 5: The overall architecture of Geometry and Question Aware Model (GQAM). Geometry-informed Visual Encoder fuses visual object features and scene text features considering their geometry relationships. Question-guided Encoder dynamically selects relevant visual features to questions. Multimodal Decoder takes multimodal features to generate a text-aware caption.

follows. For simplicity, we only show the calculation of one head attention in equations.

$$b_i = (c_i^x, c_i^y, w_i, h_i) \quad (1)$$

$$b_j = (c_j^x, c_j^y, w_j, h_j) \quad (2)$$

$$s_{ij}^g = \log\left(\frac{|c_i^x - c_j^x|}{w_i}, \frac{|c_i^y - c_j^y|}{h_i}, \frac{w_j}{w_i}, \frac{h_j}{h_i}\right) W^g, \quad (3)$$

$$s_{ij}^v = v_i W_Q^v (v_j W_K^v)^T, \quad (4)$$

$$s_{ij} = \frac{s_{ij}^g \exp s_{ij}^v}{\sum_{k=1}^N s_{ik}^g \exp s_{ik}^v}, \quad (5)$$

$$\alpha_i = [s_{i1}, s_{i2}, \dots, s_{iN}], \quad (6)$$

$$\hat{v}_i = \alpha_i V, \quad (7)$$

where (c^x, c^y, w, h) means (center coordinates, width, height) of a bounding box. W^g, W_Q^v, W_K^v are learned projection matrices. b_i and v_i are the bounding box and region feature for the i^{th} region in joint visual region sequence. s_{ij}^g and s_{ij}^v means the geometry attention score and visual attention score, respectively. α_i means the geometry informed attention weight for the i^{th} vision region. $N = N_{obj} + N_{ocr}$. \hat{v}_i is the geometry informed visual feature.

4.2 Question-guided Encoder

To satisfy multiple information needs in one caption, we propose to progressively find answers for each question from the image and then incorporate answers with the initial caption to produce a final text-aware caption. Thus, it's critical to locate corresponding visual regions for each question before the caption generation. Besides, for different words in a question, their relevant vision regions should be different. As the example shown in Figure 5, the word 'jersey' is relevant with the 'jersey' object regions and the word 'number' is relevant with the scene text region '33'. Taking into account these two points, we design a Question-guided Encoder to dynamically

select relevant visual features for each question at word level before generating text-aware captions.

With the geometry informed visual features $\hat{V} = [\hat{v}_1, \dots, \hat{v}_N]$ as the key and value, the question token embeddings $T^{que} = [t_1^{que}, t_2^{que}, \dots, t_{N^{que}}^{que}]$ as the query, the question-guided attention is calculated as:

$$s_{ij}^q = t_i^{que} W_Q^q (\hat{v}_j W_K^q)^T, \quad (8)$$

$$\beta_i = \text{Softmax}([s_{i1}^q, s_{i2}^q, \dots, s_{iN}^q]), \quad (9)$$

$$t_i^v = \beta_i \hat{V}, \quad (10)$$

where W_Q^q, W_K^q are learned projection matrices, \hat{v}_j is the geometry informed visual feature, β_i means the visual attention weight for the i^{th} token of the question, t_i^v is the attended visual feature.

To generate a caption that accurately answers a question, key words in questions can always help the decoder describe the relationship between scene texts and objects. For example, for a question 'who wrote the book', object word 'book' and author name can be fluently connected with a phrase 'written by', which can be easily inferred from the keyword 'wrote' in the question. So, besides relevant visual features, we retain the token-level text features of questions and combine them with sum operation:

$$\hat{t}_i^{que} = t_i^v W_V^q + t_i^{que} W_T^q, \quad (11)$$

where W_V^q, W_T^q are learned projection matrices, \hat{t}_i^{que} is the multimodal feature for the i^{th} token of the question.

4.3 Multimodal Decoder

Multimodal Decoder generates a text-aware caption step by step. At each time step t , Multimodal Decoder first fuses multimodal features with multiple transformer layers. The i^{th} transformer layer takes output of the last layer, $[\hat{V}_{i-1}^{obj}, \hat{V}_{i-1}^{ocr}, \hat{t}_{i-1}^{que}, T_{i-1}^{ini}, Y_{i-1}^{dec}]$, as input, where \hat{V}_0^{obj} and \hat{V}_0^{ocr} are visual features output by the Geometry-informed Visual Encoder, \hat{t}_0^{que} are token-level question features encoded by the Question-guided Encoder, T_0^{ini} are token-level features of the

Table 2: Comparison of different models on the ControlTextCaps and ControlVizwiz datasets. ‘Question’ denotes whether the model takes questions as input.

Dataset	Model	Question	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROUGE-L	CIDEr	SPICE	AnsRecall
ControlTextcaps	M4C-Captioner	✗	34.68	21.08	13.53	8.98	15.53	32.05	102.41	20.58	-
	ControlM4CC	✓	52.86	40.00	30.75	23.81	25.76	48.48	215.45	37.00	46.56
	GQAM w/o GE	✓	53.99	41.23	32.12	25.24	26.39	49.91	229.55	38.30	47.14
	GQAM	✓	54.24	41.55	32.50	25.66	26.52	50.07	231.74	38.44	50.92
ControlVizwiz	M4C-Captioner	✗	36.88	22.28	14.06	8.90	15.19	34.12	91.08	17.24	-
	ControlM4CC	✓	50.97	38.70	30.03	23.32	24.61	49.57	195.94	33.38	33.24
	GQAM w/o GE	✓	53.00	40.67	31.90	25.03	25.25	50.55	210.60	34.58	33.39
	GQAM	✓	51.61	39.62	31.06	24.33	24.82	49.73	201.35	33.81	34.62

initial caption. Y_0^{dec} are embeddings of a fixed-length list storing tokens decoded in previous times steps. With the features output by the final layer of transformers, a pointer network and a fully-connected layer are used to decode the t^{th} token as follows:

$$\hat{y}_t^{voc} = \text{FC}(z_{t-1}^{dec}), \quad (12)$$

$$\hat{y}_{t,k}^{ocr} = (W_z^{dec} z_{t-1}^{dec} + b^{dec})^T (W_z^{ocr} z_k^{ocr} + b^{ocr}), \quad (13)$$

$$\hat{y}_t^{ocr} = [\hat{y}_{t,1}^{ocr}, \hat{y}_{t,2}^{ocr}, \dots, \hat{y}_{t,N_{ocr}}^{ocr}], \quad (14)$$

$$\hat{y}_t = \text{Sigmoid}([\hat{y}_t^{voc}, \hat{y}_t^{ocr}]) \quad (15)$$

where z_{t-1}^{dec} is final-layer output of the $(t-1)^{th}$ token in previous decoded token list. z_k^{ocr} is the final-layer output of the k^{th} scene text region. $W_z^{dec}, W_z^{ocr}, b^{dec}, b^{ocr}$ are trainable parameters. \hat{y}_t^{voc} is the predicted score distributed on the fixed vocabulary. \hat{y}_t^{ocr} is the predicted score distributed on the ocr vocabulary made up by scene texts detected from the image. \hat{y}_t is the final score distributed on the joint vocabulary.

4.4 Training

Training Objective. During training, we apply the cross-entropy loss for the text-aware caption generation:

$$\text{Loss} = - \sum_{t=1}^l y_t \log(\hat{y}_t), \quad (16)$$

where y^t is the ground-truth of the t^{th} token in the target caption Y . For a target token appearing in both fixed vocabulary and ocr vocabulary or matching multiple scene text tokens, we randomly sample a possible distribution during training.

Training Strategy. Given different initial captions as input during training, the model may learn different kinds of abilities. When trained with an initial caption only containing major object words (e.g. automatic initial caption C^{ini}) as input, the model should not only improve the image object description (namely description ability) but also add accurate scene text information (namely answering ability) to reduce the generation loss. When trained with a caption similar with ground-truth caption but only lacking scene texts (e.g. pseudo initial caption \tilde{C}^{ini}), the model could focus more on the answering ability. Thus, we further explore the influence of different training strategies to these two abilities. The ‘pseudo’ or ‘auto’ means only using pseudo initial caption \tilde{C}^{ini} or automatic initial caption C^{ini} as initial captions during training, ‘rand(auto, pseudo)’ means randomly choose one of them as the initial caption for each instance during training. Note during inference, we only

use automatic initial caption C^{ini} as initial captions because in the real use cases, initial captions are not always in high quality.

5 EXPERIMENTS

In this section, we carry out extensive experiments to evaluate the proposed model on the two constructed Qc-TextCap datasets.

5.1 Experimental setup

Baselines. 1) *Non-controllable baseline: M4C-Captioner* [24], which is the state-of-the-art text-aware image captioning model. It fuses object features and scene text features with multi-layer transformers and decodes a caption with a fully connected layer and pointer network. 2) *Controllable baseline: ControlM4CC*, which extends the M4C-Captioner into a question-controlled captioning model. It has identical architecture with M4C-Captioner, but concatenates initial caption features and question features with object features and scene text features as input. 3) *Controllable variant of GQAM: GQAM w/o GE*, which drops the Geometry-informed Visual Encoder in GQAM to evaluate contributions from different components in the final GQAM model.

Evaluation Metrics. We use the common captioning evaluation metrics BLEU [20], METEOR [7], ROUGE-L [16], CIDEr [27] and SPICE [1] to measure the overall quality of generated captions. Among these metrics, CIDEr is most suitable for text-aware captioning because it puts more importance on rarer words, especially scene texts. To specifically evaluate the question answering ability of question-controlled models, we further calculate the recall of answer tokens as AnsRecall.

Implementation Details. On both datasets, we set the max length of the initial caption and the concatenated question as 20. The max length of generated text-aware caption is set to 30. We extract 100 object bounding boxes and at most 50 scene text bounding boxes for each image. During training, we set batch size as 50 for both datasets. Training steps are set as 10,000 and 16,000 for ControlVizWiz and ControlTextCaps, respectively. During inference, we utilize greedy search to decode captions if not specified.

5.2 Qc-TextCap Performance Evaluation

We first conduct ablation studies of different models on captioning performance for the Qc-TextCap task.

Comparison of non-controllable and controllable models. We train all models with the ‘auto’ training strategy to fairly compare different text-aware captioning models. As M4C-Captioner is a non-controllable baseline without taking the question as input,

Table 3: Comparison of different training strategies. ‘pseudo’ or ‘auto’ means only using pseudo initial captions \tilde{C}^{ini} or automatic initial captions C^{ini} as initial captions during training, respectively. ‘rand(pseudo, auto)’ means randomly choosing one of them for each training sample. During inference, only automatic initial captions are used as initial captions.

Dataset	Model	train strategy	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	ROUGE-L	CIDEr	SPICE	AnsRecall
ControlTextcaps	GQAM	auto	54.24	41.55	32.50	25.66	26.52	50.07	231.74	38.44	50.92
		pseudo	43.26	29.39	20.74	14.72	19.89	38.97	143.36	25.46	49.47
		rand(auto, pseudo)	54.63	42.01	32.96	26.13	26.83	50.50	238.20	38.69	51.27
ControlVizwiz	GQAM	auto	53.00	40.67	31.90	25.03	25.25	50.55	210.60	34.58	33.39
		pseudo	44.85	30.56	21.70	15.67	20.01	41.60	140.08	23.77	34.70
		rand(auto, pseudo)	54.41	42.43	33.64	26.79	25.98	51.65	223.23	35.85	33.72

we do not calculate the AnsRecall metric of the model. As shown in Table 2, with questions as guidance, question-controlled models achieve much better captioning performance than M4C-Captioner. Among question-controlled models, our model GQAM outperforms ControlM4CC in both caption generation and question answering, which demonstrates the effectiveness of the overall architecture. In both datasets, compared with ControlM4CC, GQAM w/o GE achieves significant improvement, especially on CIDEr scores (+14.1/+13.8). This shows that Question-guided Encoder helps locate relevant scene text regions and is critical for describing the relationship between scene texts and objects during generation. GQAM achieves better AnsRecall scores than GQAM w/o GE on both datasets, which shows Geometry-informed Visual Encoder indeed helps generate more scene texts by fusing visual region features that are spatially related. As for captioning performance, GQAM outperforms GQAM w/o GE on ControlTextCaps but underperforms GQAM w/o GE on ControlVizWiz. This is because that bounding boxes on ControlVizWiz are obviously worse than ControlTextCaps due to the image quality issues. Geometry information of bounding boxes plays a crucial role in Geometry-informed Visual Encoder, so inaccurate bounding boxes can introduce noise into visual features and result in generating some irrelevant scene texts. For simplicity, we use GQAM to refer to GQAM w/o GE in the following experiments on ControlVizWiz dataset.

Comparison of different training strategies. Based on the best performed model on each dataset, we further explore the influence of ‘auto’, ‘pseudo’ and ‘rand(auto, pseudo)’ training strategies. Table 3 presents the comparison results. We find that on test sets of both ControlTextCaps and ControlVizWiz, models with ‘pseudo’ training strategy achieve similar question answering performance to the ones with ‘auto’ training strategy, but achieve much worse captioning performance. This proves that only using pseudo initial captions is not good for the description ability of models. Besides, compared with the ‘auto’ training strategy, the ‘rand(auto, pseudo)’ training strategy could further improve both captioning scores and question answering scores on both datasets. This shows that making the model only focus on adding scene text information in part of training steps could strengthen the answering ability.

We provide a more detailed ablation study about the contribution of each modality in our supplementary material.

5.3 Diversity Evaluation

Our model is able to generate diverse personalized captions for an image given different questions. To measure the diversity of these captions, we choose widely used Div- n [3, 5, 8] and SelfCIDEr [30]

Table 4: Diversity evaluation of our GQAM and the text-aware captioning model M4C-Captioner.

Dataset	Model	Div-1	Div-2	SelfCIDEr
ControlTextCaps	M4C-Captioner	7.44	21.11	62.58
	GQAM	14.72	38.00	78.32
ControlVizWiz	M4C-Captioner	6.41	19.97	56.36
	GQAM	10.88	28.71	63.06

Table 5: Human evaluation of accurate scene text information (ST Info) and overall caption quality. For simplicity, we use M4CC to refer to M4C-Captioner

Dataset		ST Info	Overall Quality
ControlTextcaps	GQAM>M4CC	43.48%	51.38%
	GQAM≈M4CC	42.29%	27.67%
	GQAM<M4CC	14.23%	20.95%
ControlVizWiz	GQAM>M4CC	44.30%	41.77%
	GQAM≈M4CC	39.24%	24.05%
	GQAM<M4CC	16.46%	34.18%

as diversity metrics. Div- n is the ratio of distinct n -gram to the total number of words in the multiple captions for an image. SelfCIDEr is calculated by applying latent semantic analysis on a CIDEr score matrix, which is computed among a set of captions for each image.

For each image, GQAM generates multiple captions with different questions from a relevant question set. For the text-aware image captioning model M4C-captioner, we apply beam decoding to get the same number of different captions. As shown in Table 4, our GQAM significantly outperforms M4C-Captioner on all diversity metrics on both datasets. This indicates that our question control signals indeed guide the model to focus on different image regions and generate personalized captions.

5.4 Qualitative Evaluation

To verify question control signals’ contribution in generating more informative captions, we ask 6 human evaluators to evaluate the scene text information and the overall quality of the text-aware caption. For each image, given a caption generated by M4C-Captioner and a caption generated by GQAM, human evaluators are asked to choose 1) which one conveys more scene text information that is accurately described with the object; 2) which one has better overall quality. As shown in Table 5, our question-controlled model GQAM accurately describes more scene texts than M4C-Captioner. Besides, GQAM also performs better in overall quality. This indicates that



M4C-Captioner:
a book cover with the title **uniunea europeana**

Ground A: a book by **gabriela carmen pascariu** about **uniunea europeana**



M4C-Captioner:
a bottle of the **royal legac** sits on a table

Ground A: a bottle of **Royal Legacy malt whiskey** and the box it came in.

Ground B: bottle of alcohol that says **The Royal Legacy** by a green box.



M4C-Captioner:
a phone that has the word mil. at&t on it

Ground A: white phone with a screen that says **August 12th** on it.

Ground B: a mobile phone using **AT&T's** network shows an app on its screen that is used to monitor baby **feeding** times and amounts.

- (a) **Initial caption:** a book
-----**question-controlled text-aware captions**-----
Questions A: **what is the title of the book? who wrote the book?**
ControlM4CC: a book by **gabriela carmen pascariu** called **uniunea europeana**
GQAM: a book by **gabriela carmen pascariu** called **uniunea politici si pietele europeneana**
- (b) **Initial Caption:** a bottle next to a box
-----**question-controlled text-aware captions**-----
Questions A: **what is the brand on the bottle? what is in the bottle?**
ControlM4CC: a bottle of **royal royal legac** next to a box of it
GQAM: a bottle of **royal legac malt whisky liqueur** next to a box of the new Orleans
Questions B: **what does the label on the bottle say? what is in the bottle?**
ControlM4CC: a bottle of alcohol that says the **royal legac** on it
GQAM: a bottle of beer that says " **royal legacy** " is on the table
- (c) **Initial Caption:** a white phone on a wooden table
-----**question-controlled text-aware captions**-----
Questions A: **what is the date shown on the phone?**
ControlM4CC: a white phone on a table that says ' **sunday august 12 2012** ' on it
GQAM: a phone on a wooden table with the date of **sunday august 12 2012**
Questions B: **what app is installed on the phone?**
ControlM4CC: a phone with the app **feeding** on the screen
GQAM: a phone with the app **feeding** and **sleeping** on the screen

Figure 6: Qualitative results of M4C-Captioner, ControlM4CC and GQAM.

besides conveying more scene text information, GQAM is also good at organizing them together in natural language.

We show some examples from test sets in Figure 6. First, general captioning model M4C-Captioner cannot make full use of multiple scene text information in an image. But with multiple questions as control signals, question-controlled text-aware models are guided to focus on multiple scene text regions. For example, as shown in Figure 6(a), M4C-Captioner only describes the book title without the author name. For question-controlled models, either ControlM4CC or our GQAM successfully outputs these two parts of information given target questions. Second, with different questions as control signals, question-controlled models could pay attention to different scene text regions and generate personalized captions. As shown in Figure 6(b), when asked about the brand on a bottle, GQAM focuses on the 'sunday august 12 2012' region, and when asked about the installed application, it describes app names with other scene texts. Third, our model GQAM could find relevant scene text regions better than ControlM4CC. In the example presented in Figure 6(b), given questions about the brand and the content of a bottle, ControlM4CC only describes the brand but GQAM answers both two questions in one caption with the help of the Question-guided Encoder. Further, we find that without background knowledge, GQAM may misunderstand some scene text information. As shown in Figure 6(c), when asked about the application name on a phone, GQAM focuses on two scene text regions: 'feeding' and 'sleeping'. These two regions are very similar with applications in shape but are actually two functions of an application according to the text information and background knowledge. More qualitative examples can be found in our supplementary material.

6 CONCLUSION

To generate personalized text-aware captions for visually impaired people, we propose a new challenging task named Question-controlled Text-aware Image Captioning (Qc-TextCap). We use questions about scene texts to control text-aware caption generation due to its convenience in interaction. The Qc-TextCap task requires models to comprehend questions, find relevant scene text regions and incorporate answers with an initial caption to produce a final text-aware caption. To support this task, we automatically construct datasets ControlTextCaps and ControlVizWiz based on existing text-aware captioning datasets, which will be released publicly. We further propose a Geometry and Question Aware Model (GQAM) to progressively encodes relevant visual features and text features. On both datasets, GQAM achieves better performance than carefully designed baselines on both captioning and question answering metrics. We further prove that the model with questions as control signals can generate more informative and diverse captions.

ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China (No. 61772535 and No. 62072462), Beijing Natural Science Foundation (No. 4192028), National Key R&D Program of China (No. 2020AAA0108600).

REFERENCES

- [1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. SPICE: Semantic Propositional Image Caption Evaluation. In *ECCV (5) (Lecture Notes in Computer Science, Vol. 9909)*. Springer, 382–398.
- [2] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *CVPR*. IEEE Computer Society, 6077–6086.

- [3] Jyoti Aneja, Harsh Agrawal, Dhruv Batra, and Alexander G. Schwing. 2019. Sequential Latent Spaces for Modeling the Intention During Diverse Image Captioning. In *ICCV*. IEEE, 4260–4269.
- [4] Ali Furkan Biten, Rubèn Tito, Andrés Mafra, Lluís Gómez i Bigorda, Marçal Rusiñol, C. V. Jawahar, Ernest Valveny, and Dimosthenis Karatzas. 2019. Scene Text Visual Question Answering. In *ICCV*. IEEE, 4290–4300.
- [5] Shizhe Chen, Qin Jin, Peng Wang, and Qi Wu. 2020. Say As You Wish: Fine-Grained Control of Image Caption Generation With Abstract Scene Graphs. In *CVPR*. IEEE, 9959–9968.
- [6] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. 2019. Show, Control and Tell: A Framework for Generating Controllable and Grounded Captions. In *CVPR*. Computer Vision Foundation / IEEE, 8307–8316.
- [7] Michael J. Denkowski and Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *WMT@ACL*. The Association for Computational Linguistics, 376–380.
- [8] Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander G. Schwing, and David A. Forsyth. 2019. Fast, Diverse and Accurate Image Captioning Guided by Part-Of-Speech. In *CVPR*. Computer Vision Foundation / IEEE, 10695–10704.
- [9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*. Association for Computational Linguistics, 4171–4186.
- [10] Adam Fisch, Kenton Lee, Ming-Wei Chang, Jonathan H. Clark, and Regina Barzilay. 2020. CapWAP: Image Captioning with a Purpose. In *EMNLP (1)*. Association for Computational Linguistics, 8755–8768.
- [11] Danna Gurari, Yanan Zhao, Meng Zhang, and Nilavra Bhattacharya. 2020. Captioning Images Taken by People Who Are Blind. In *ECCV (17) (Lecture Notes in Computer Science, Vol. 12362)*. Springer, 417–434.
- [12] Simao Herdade, Armin Kappeler, Kofi Boakye, and Joao Soares. 2019. Image Captioning: Transforming Objects into Words. In *NeurIPS*. 11135–11145.
- [13] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. 2020. Iterative Answer Prediction With Pointer-Augmented Multimodal Transformers for TextVQA. In *CVPR*. IEEE, 9989–9999.
- [14] Lun Huang, Wenmin Wang, Jie Chen, and Xiaoyong Wei. 2019. Attention on Attention for Image Captioning. In *ICCV*. IEEE, 4633–4642.
- [15] Yash Kant, Dhruv Batra, Peter Anderson, Alexander G. Schwing, Devi Parikh, Jiasen Lu, and Harsh Agrawal. 2020. Spatially Aware Multimodal Transformers for TextVQA. In *ECCV (9) (Lecture Notes in Computer Science, Vol. 12354)*. Springer, 715–732.
- [16] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.
- [17] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *ECCV (5) (Lecture Notes in Computer Science, Vol. 8693)*. Springer, 740–755.
- [18] Jiasen Lu, Caiming Xiong, Devi Parikh, and Richard Socher. 2017. Knowing When to Look: Adaptive Attention via a Visual Sentinel for Image Captioning. In *CVPR*. IEEE Computer Society, 3242–3250.
- [19] Meredith Ringel Morris, Jazette Johnson, Cynthia L. Bennett, and Edward Cutrell. 2018. Rich Representations of Visual Content for Screen Reader Users. In *CHI*. ACM, 59.
- [20] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL*. ACL, 311–318.
- [21] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67.
- [22] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100, 000+ Questions for Machine Comprehension of Text. In *EMNLP*. The Association for Computational Linguistics, 2383–2392.
- [23] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7008–7024.
- [24] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. 2020. TextCaps: A Dataset for Image Captioning with Reading Comprehension. In *ECCV (2) (Lecture Notes in Computer Science, Vol. 12347)*. Springer, 742–758.
- [25] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards VQA Models That Can Read. In *CVPR*. Computer Vision Foundation / IEEE, 8317–8326.
- [26] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*. 5998–6008.
- [27] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. 2015. CIDEr: Consensus-based image description evaluation. In *CVPR*. IEEE Computer Society, 4566–4575.
- [28] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *CVPR*. IEEE Computer Society, 3156–3164.
- [29] Jing Wang, Jinhui Tang, and Jiebo Luo. 2020. Multimodal Attention with Image Text Spatial Relationship for OCR-Based Image Captioning. In *ACM Multimedia*. ACM, 4337–4345.
- [30] Qingzhong Wang and Antoni B. Chan. 2019. Describing Like Humans: On Diversity in Image Captioning. In *CVPR*. Computer Vision Foundation / IEEE, 4195–4203.
- [31] Zhaokai Wang, Renda Bao, Qi Wu, and Si Liu. 2021. Confidence-aware Non-repetitive Multimodal Transformers for TextCaps. In *AAAI*. AAAI Press, 2835–2843.
- [32] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML (JMLR Workshop and Conference Proceedings, Vol. 37)*. JMLR.org, 2048–2057.
- [33] Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei A. F. Florêncio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. 2020. TAP: Text-Aware Pre-training for Text-VQA and Text-Caption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8751–8761.
- [34] Yue Zheng, Yali Li, and Shengjin Wang. 2019. Intention Oriented Image Captions With Guiding Objects. In *CVPR*. Computer Vision Foundation / IEEE, 8395–8404.
- [35] Qi Zhu, Chenyu Gao, Peng Wang, and Qi Wu. 2021. Simple is not Easy: A Simple Strong Baseline for TextVQA and TextCaps. In *AAAI*. AAAI Press, 3608–3615.

The sign that says **Beacon Lighting** on it above the entrance to a store .

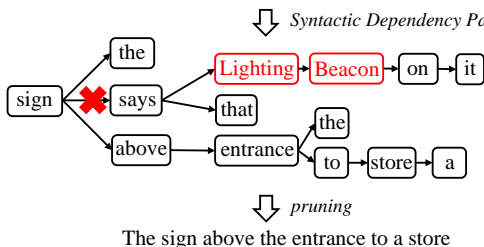


Figure 7: Removing scene texts in a text-aware caption with Syntactic Dependency Parsing.

A ADDITIONAL DATASET CONSTRUCTION DETAILS

A.1 Scene Text Removing

As shown in Figure 7, ‘Beacon Lighting’ are detected scene texts in the text-aware caption. We then perform syntactic dependency parsing to get a dependency tree of the caption. By removing the branch containing scene texts ‘Beacon’ and ‘Lighting’, we get the pseudo initial caption ‘The sign above the entrance to a store’.

A.2 Question Filtering

Questions generated by the pre-trained question generation model are not all about scene texts. For example, for the caption ‘The sign that says Beacon Lighting on it above the entrance to a store’, the model generates a question ‘where is the sign’, which is irrelevant with the scene text information. We drop this kind of question mainly by judging whether their answers could be found in OCR results. Due to the imperfect of OCR system, a phrase of scene texts may be recognized as multiple OCR tokens. So we keep a question if any token in its answer could be found in OCR results. After the question filtering, only text-aware captions with at least one question towards scene texts are included in our question-controlled datasets.

A.3 Question Cleaning

For a text-aware caption containing multiple scene texts, a question asked for one scene text may contain scene text answers of other questions. For example, for a text-aware caption ‘A book by Mark Hillary titled CEO sits on top of a box’, a generated question ‘what is the title of the book by Mark Hillary’ contains the answer of another question ‘who wrote the book’. Using this kind of questions will leak answers for other questions to the model, so we further remove scene text information in this kind of questions with the simplifying rules introduced in *Initial Caption Generation*.

Besides redundant scene texts, questions generated by the model also contain some extra description information. For a target caption ‘the sign that says Beacon Lighting on it above the entrance’, the question ‘what does the sign above the entrance say on it’ provides extra description information ‘above the entrance’ that is not in the initial caption ‘yellow sign’. To remove extra description information, we design a template conversion step and backbone extraction step. At template conversion step, We first collect clean

Table 6: Statistics of training images and captions in Qc-TextCap datasets and text-aware captioning datasets

Dataset	Image	Caption
ControlTextCaps	20,217	65,192
TextCaps	22,101	104,354
ControlVizWiz	10,763	25,339
VizWiz-Caption	28,706	121,704

questions of high frequency for popular objects, namely template questions, such as ‘what is the title of the book’ and ‘what is the author of the book’ for object ‘book’. Then we align raw questions to template questions by keywords. For example, if ‘what book’ appears in a raw question, we convert the question to ‘what is the title of the book’. For questions that are not cleaned by template conversion, we apply a backbone extraction step to remove modifiers by Syntactic Dependency Parsing.

A.4 Training of AoANet

To generate appropriate initial captions, we train an in-domain general captioning model AoANet given image I and pseudo initial caption \tilde{C}^{ini} pairs. For all captions in raw text-aware captioning datasets (TextCaps/VizWiz-Caption), we produce pseudo initial captions but only part of them are included in our Qc-TextCap datasets due to the question filter, as shown in Table 6. Only using training data of Qc-TextCap dataset (ControlTextCaps/ControlVizWiz) to train an AoANet model will cause an obvious gap in captioning performance between training split and test split. To alleviate the gap problem, we train the AoANet with all image and pseudo initial caption pairs in TextCaps/VizWiz-Caption training split as training data. As shown in Table 7, on both datasets, AoANet trained with extra data achieves worse captioning performance on train splits of Qc-TextCap datasets but better captioning performance on test splits. Especially, AoANet trained with extra data achieves comparable object precision with the one trained with ControlTextCaps or ControlVizWiz on the training split and better object precision on the test split.

B ADDITIONAL ABLATION STUDY

We perform a more detailed ablation study to show the importance of each modality for Qc-TextCap task. As shown in Table 8, all modalities of input are necessary for GQAM. First and foremost, as the control signal, the question contributes most to the captioning performance. Second, by comparing GQAM w/o ini with GQAM trained by ‘rand(auto, pseudo)’ strategy, we find the initial captions is also important in Qc-TextCap task. Finally, to select scene text more accurately, it’s necessary to integrate scene text region features with other features.

C ADDITIONAL QUALITATIVE EXAMPLES

We present more qualitative examples in Figure 8. As shown in Figure 8(a) and Figure 8(b), our GQAM express more scene text information than M4C-Captioner with multiple questions as control signals. Examples in Figure 8(c) and 8(e) show that, with different questions, question-controlled models ControlM4CC and

Table 7: Image captioning performance of AoANet models trained on different training data. $p^{obj}(C^{ini})$ means the object precision of generated initial captions. These metrics are all computed at image level. ‘Split’ refers to the train or test split in ControlTextCaps (ControlVizWiz).

Dataset	Train Data	Split	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	Rouge-L	CIDEr	SPICE	$p^{obj}(C^{ini})$
ControlTextcaps	ControlTextcaps	train	34.13	28.47	23.34	19.20	17.44	42.83	72.17	19.88	96.43
		test	24.57	17.1	11.51	7.72	11.43	31.14	26.96	11.82	81.13
	Textcaps	Train	35.96	28.36	21.72	16.47	16.11	39.99	51.84	17.56	94.40
		test	30.3	21.41	14.56	9.70	12.41	32.65	31.18	12.63	82.04
ControlVizWiz	ControlVizWiz	train	29.59	25.30	21.67	18.64	18.05	45.09	108.43	24.20	92.30
		test	16.57	10.67	6.84	4.54	8.64	27.14	23.58	9.19	53.64
	VizWiz-Caption	train	28.00	20.11	14.10	9.78	13.07	35.46	46.05	15.53	82.25
		test	25.70	16.90	10.93	7.05	10.66	31.00	33.26	11.16	66.51

Table 8: Addition Ablation Study. ‘que’, ‘ini’, ‘ocr’ and ‘obj’ refer to the question features T^{que} , initial caption features T^{ini} , scene text region features V^{ocr} and object region features V^{obj} respectively. Note ‘w/o ocr’ means scene text region features V^{ocr} is not used to fuse with other features but is still used in the pointer network.

Dataset	Model	Training Strategy	BLEU1	BLEU2	BLEU3	BLEU4	METEOR	Rouge-L	CIDEr	SPICE
ControlTextcaps	GQAM w/o que	auto	37.40	23.10	14.99	9.96	16.91	33.02	109.69	21.03
	GQAM w/o ocr	auto	53.25	40.53	31.55	24.79	25.72	49.10	221.20	37.34
	GQAM w/o obj	auto	53.95	41.30	32.23	25.38	26.38	49.72	230.11	38.13
	GQAM w/o ini	-	54.60	41.80	32.55	25.59	26.74	50.42	232.66	38.78
	GQAM	auto	54.24	41.55	32.50	25.66	26.52	50.07	231.74	38.44
	GQAM	rand(auto, pseudo)	54.63	42.01	32.96	26.13	26.83	50.50	238.20	38.69
ControlVizWiz	GQAM w/o que	auto	36.74	22.51	14.25	9.00	15.39	34.38	93.25	18.42
	GQAM w/o ocr	auto	50.57	38.48	29.76	23.00	23.22	48.82	187.54	31.99
	GQAM w/o obj	auto	51.99	39.65	30.95	24.17	24.69	49.66	203.67	34.32
	GQAM w/o ini	-	51.84	39.44	30.71	23.99	25.26	50.20	203.29	33.99
	GQAM	auto	53.00	40.67	31.90	25.03	25.25	50.55	210.60	34.58
	GQAM	rand(auto, pseudo)	54.41	42.43	33.64	26.79	25.98	51.65	223.23	35.85

GQAM generate personalized text-aware captions. In Figure 8(c), compared with ControlM4CC, GQAM concatenates the main title and subtitle scene texts to answer the second question, which indicates Geometry-informed Visual Encoder indeed helps recombine a scene text phrase that is separated due to typesetting. Figure 8(d) shows the contribution of a good initial caption. M4C-Captioner describes one of scene texts but neglects the type of the sign (‘highway sign’), which is important to understand this image. With a

good initial caption as input, both ControlM4CC and GQAM retain this key information in their generated captions. Figure 8(f) presents that the question could help model to understand the relationship between objects and scene texts. M4C-Captioner just treats ‘rackspace’ as a word on the sign, but GQAM and ControlM4CC understand that it’s a company which the sign is advertising for.



M4C-Captioner:
a can of london pride beer is on a table

Ground A: a can of Fuller's London pride sits on a table.

Initial caption: a can

question-controlled text-aware captions:

Questions A: what is the brand on the can? what is in the can?

ControlM4CC: a can of london pride beer is on a table

GQAM: a can of fuller's 's pride premium beer on a table

(a)



M4C-Captioner:
a license plate that says monyman on it

Ground A: virginia license plate with MONYMAN on it that expires in feb 2011.

Initial Caption: a license plate

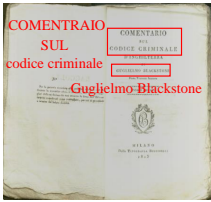
question-controlled text-aware captions:

Questions A: when does the license plate it expire? which state does the plate belong to? what is on the license plate that expires?

ControlM4CC: a virginia license plate that is on a car

GQAM: a virginia license plate that has the word monyman on it

(b)



M4C-Captioner:
a book is open to page that says comentario

Ground A: open book on a page that says COMMENTARIO on the top.

Ground B: an old book displaying the title Comentario written by Guglielmo Blackstone

Initial Caption: a book is open to a page

question-controlled text-aware captions:

Questions A: what is the title of the book?

ControlM4CC: an open book with the title comentario on the top

GQAM: a book is open to a page titled comentario

Questions B: who is the author of the book? what is the title of the book?

ControlM4CC: an open book with the title comentario di guglielmo blackstone

GQAM: a book titled comentario codice criminale by guglielmo blackstone

(c)



M4C-Captioner:
a sign that says 'al khawaneej' on it

Ground A: a highway direction sign showing Al Khawaneej to the left, Hatta straight ahead, and Dubai Academic City to the right.

Initial Caption: a highway sign

question-controlled text-aware captions:

Questions A: what does the sign show? what does the sign show?

ControlM4CC: a highway sign showing hatta oman is to the right

GQAM: a highway sign showing al khawaneej and hatta oman

(d)



M4C-Captioner:
a player with the number 23 on his jersey

Ground A: a picture of two WNBA players number 14 from Connecticut and number 23 from the Lynx.

Ground B: a female basketball player from the team Lynx

Initial Caption: a basketball player wearing a jersey

question-controlled text-aware captions:

Questions A: what is the number of the player?

ControlM4CC: a basketball player wearing a white jersey with the number 23 on it

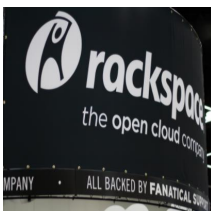
GQAM: a basketball player wearing the number 23 on his shirt

Questions B: which team does the player belong to?

ControlM4CC: a basketball player wearing a jersey that says lynx on it

GQAM: a basketball player for the lynx is standing in front of a basketball game

(e)



M4C-Captioner:
a sign that says 'rackspace' on it

Ground A: a sign for rackspace, the open cloud company.

Initial Caption: a black sign

question-controlled text-aware captions:

Questions A: what is the sign advertising for?

ControlM4CC: a black sign for rackspace displays a large black background

GQAM: a black sign for rackspace the open cloud company in the center

(f)

Figure 8: Qualitative Results of M4C-Captioner, ControlM4CC and GQAM.