Towards Accurate Text-based Image Captioning with Content Diversity Exploration

Guanghui Xu[1,2]*, Shuaicheng Niu[1]*, Mingkui Tan[1,4], Yucheng Luo[1], Qing Du[1,4]†, Qi Wu[3]
[1]South China University of Technology, [2]Pazhou Laboratory, [3] University of Adelaide
[4]Key Laboratory of Big Data and Intelligent Robot, Ministry of Education
sexuguanghui@mail.scut.edu.cn, {mingkuitan, duqing}@scut.edu.cn, qi.wu01@adelaide.edu.au

Abstract

Text-based image captioning (TextCap) which aims to read and reason images with texts is crucial for a machine to understand a detailed and complex scene environment, considering that texts are omnipresent in daily life. This task, however, is very challenging because an image often contains complex texts and visual information that is hard to be described comprehensively. Existing methods attempt to extend the traditional image captioning methods to solve this task, which focus on describing the overall scene of images by one global caption. This is infeasible because the complex text and visual information cannot be described well within one caption. To resolve this difficulty, we seek to generate multiple captions that accurately describe different parts of an image in detail. To achieve this purpose, there are three key challenges: 1) it is hard to decide which parts of the texts of images to copy or paraphrase; 2) it is non-trivial to capture the complex relationship between diverse texts in an image; 3) how to generate multiple captions with diverse content is still an open problem. To conquer these, we propose a novel Anchor-Captioner method. Specifically, we first find the important tokens which are supposed to be paid more attention to and consider them as anchors. Then, for each chosen anchor, we group its relevant texts to construct the corresponding anchor-centred graph (ACG). Last, based on different ACGs, we conduct the multi-view caption generation to improve the content diversity of generated captions. Experimental results show that our method not only achieves SOTA performance but also generates diverse captions to describe images.

1. Introduction

The texts are omnipresent in our daily life and play an important role in helping humans or intelligent robots to understand the physical world [13]. In the image captioning area, the texts contained in images are also of critical importance and often provide valuable information [5, 19, 20, 34, 41] for caption generation. In this sense, Sidorov *et al.* [40] propose a fine-grained image captioning task, i.e., text-based image captioning (TextCap), aiming to generate image captions that not only 'describe' visual contents but also 'read' the texts in images, such as billboards, road signs, commodity prices and etc. This task is very practical since the fine-grained image captions with rich text information can aid visually impaired people to comprehensively understand their surroundings [13]

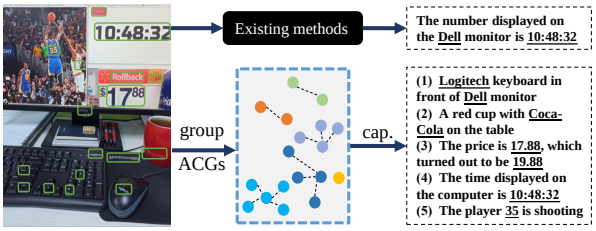

Figure 1. Comparison with existing methods. For a given image, existing methods tend to generate only one global caption. Unlike them, we first select and group texts to anchor-centred graphs (ACGs), and then decide which parts of the texts to copy or paraphrase. Our method is able to achieve higher accuracy and generate diverse captions to describe the image from different views.

Some preliminary tries for the TextCap task seek to directly extend existing image captioning methods [2, 19, 21] to this new setting. However, such methods usually tend to describe prominent visual objects or overall scenes without considering the texts in images. Recently, M4C-Captioner [40] tries to use additional OCR tools [4, 6, 31] to recognise texts in images. It is still hard to well describe the complex text and visual information within one caption. To resolve this difficulty, we propose to generate multiple diverse captions focusing on describing different parts of an image. However, there are still some challenges.

First, it is hard to decide which parts of the texts are most

*Authors contributed equally.
†Corresponding author

crucial for describing the images. As shown in Figure 1, an image often contains a lot of texts, but only a small part of the texts play a key role in caption generation. For example, *a PC keyboard contains many letters*, but we do not need a caption that covers all the recognised letters.

Second, it is non-trivial to capture the complex relationship between diverse texts in an image. The correct understanding of such a relationship is essential for generating accurate captions. For example, *to accurately describe a cup, we might use its brand and capacity*. But these texts have no relevance to the contents on the computer screen.

More critically, how to generate multiple captions describing different contents remains unknown. Current image captioning methods [2, 19, 21] often only generate a content-monotone caption. They tend to focus on a small part of the contents in the image, *such as the time in the monitor in Figure 1*. To comprehensively describe an image, one better solution is to generate diverse captions, where each caption focuses on describing one relevant part.

To address the above issues, we design a new Anchor-Captioner architecture that mainly consists of two key modules, i.e., an anchor proposal module (AnPM) and an anchor captioning module (AnCM). Specifically, AnPM is proposed to understand the texts in an image and to capture the complex relationships between different texts. To be specific, we first employ an anchor predictor to rank the importance of each token. Then, we choose several important tokens to decide which parts of texts are most informative and need to be carefully considered. After that, considering each chosen token as an anchor, we use a recurrent neural network to model its complex relationships with other tokens and to construct an anchor-centred graph (ACG) for each anchor. Each ACG denotes a group of the relevant tokens which are supposed to be included in the same caption. Based on the different ACGs for an image, we apply AnCM to generate diverse captions that cover various OCR tokens. To be specific, we first generate a visual-specific caption to model global visual information. Then, we take each ACG as guidance to refine the visual caption and generate multiple text-specific captions that contain fine-grained text information. Extensive experimental results on TextCaps dataset demonstrate the effectiveness of our proposed method.

In summary, our main contributions are as follows:

1. We propose to exploit fine-grained texts information to generate multiple captions that describe different parts of images, instead of generating a single caption to handle them as a whole.

2. We propose an anchor proposal module (AnPM) and an anchor captioning module (AnCM) to select and group texts to anchor-centred graphs (ACGs) and then generate diverse captions based on ACGs.

3. We achieve the state-of-the-art results on TextCaps dataset, in terms of both accuracy and diversity.

2. Related work

Image captioning aims to automatically generate textual descriptions of an image, which is an important and complex problem since it combines two major artificial intelligence fields: natural language processing and computer vision. Most image captioning models [2, 15, 42, 44, 45, 49] use CNNs to encode visual features and apply RNNs as language decoder to generate descriptions. Some works [16, 25, 29, 48] propose to further refine the generated sentences with multiple decoding passes. NBT [32] first generates a template without specifics and then fills it with 'object' words. RL-based methods [18, 30, 35, 39] model the sequence generation as Markov Decision Process [47] and directly maximise the metric scores.

To generate diverse image captions, many works try to control the generation in terms of style and contents. The style controllable methods [14, 17, 33] usually require additional annotations for training, such as a pair of labelled captions with different styles. Other parallel works focus on controlling the contents of the generated captions. Johnson *et al.* [22] are the first to propose the dense captioning task to describe the visual objects in a sub-region [50]. Signal-based methods [7, 8, 9, 11] sample different predictions based on the control signals to obtain diverse image captions. Our work can be seen as text-based dense captioning and aims to generate multi-view captions.

Text-based image captioning aims to generate captions describing both the visual objects and written texts. Intuitively, the text information is important for us to understand the image contents. However, the existing image captioning datasets [24, 28] have a bias that only describes the salient visual objects in the image while ignoring the text information. As a result, most image captioning models [2, 15, 42, 44, 49] unable to 'read' the texts since they don't pay attention to improve such ability. In this sense, Sidorov *et al.* [40] introduce a novel dataset, namely TextCaps, which requires a captioning model not only to 'watch' visual contents but also 'read' the texts in images. They introduce a benchmark M4C-Captioner [40], which is simply extended from the M4C [19] (for TextVQA). Specifically, they feed all the detected texts and visual contents into their captioning model to generate a global caption for an input image. However, it is difficult for a single caption to cover all the multimodal information, and the overlooked parts may be the information that people are interested in.

Different from existing methods, we propose an anchor proposal module to understand the relationship within OCR tokens and group them to construct anchor-centred graphs (ACGs). With the help of ACGs, our method is able to better describe the input image by generating diverse captions.

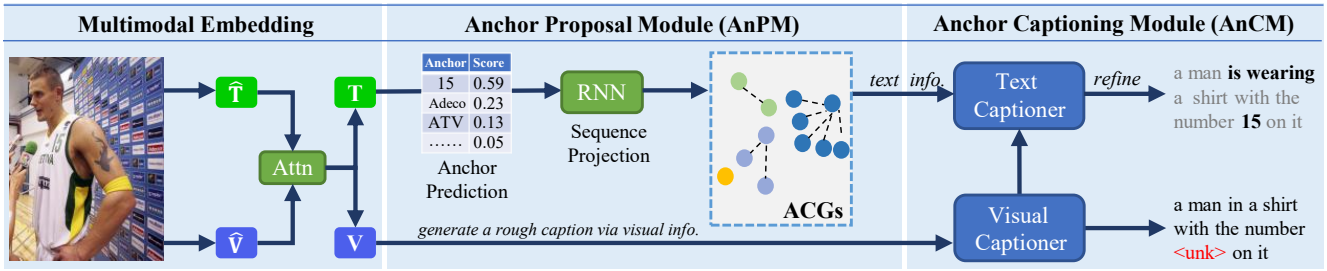

Figure 2. An illustration of Anchor-Captioner. Given an input image, (1) we first extract text and visual features $(\widehat{\mathbf{T}}, \widehat{\mathbf{V}})$ independently. Then, we fuse them to obtain multimodal features $(\mathbf{T}, \mathbf{V})$ via self-attention; (2) Following that, AnPM chooses a series of anchor-tokens based on the anchor predictions and then groups the relevant tokens by constructing anchor-centred graphs (ACGs). (3) Lastly, AnCM employs a visual-captioner to output a global visual-specific caption, and then uses a text-captioner to generate multiple text-specific captions based on the above global caption and ACGs. In this figure, we only show the generated caption with the ACG of the top-1 score.

3. Proposed method

We study text-based image captioning (TextCap) which aims to read and reason an image with texts to generate detailed captions. This task is very challenging because it is difficult to comprehensively describe images with rich information. Existing methods tend to generate one global caption that tries to describe complex contents in an image. However, such methods are unfeasible when the image contains a large number of redundant visual objects and diverse texts. To accurately describe an image, one better solution is to generate multiple captions from different views. However, several challenges still exist. First, it is hard to decide which parts of the texts of images to copy or paraphrase when images contain a lot of texts. Second, it is non-trivial to exploit the correct relationship between diverse texts in an image, which, however, is essential to accurately describe the image. More critically, how to generate multiple captions from different views for comprehensively describing images is still unknown.

In this paper, we propose a new captioning method Anchor-Captioner that aims to accurately describe images by using content diversity exploration. As shown in Figure 2, Anchor-Captioner has two main components, i.e., an anchor proposal module (AnPM) and an anchor captioning module (AnCM). AnPM chooses important texts as anchors and constructs anchor-centred graphs (ACGs) to model complex relationship between texts. AnCM takes different ACGs as input to generate multiple captions that describe different parts of an image. In this way, our method is able to choose important relevant texts to describe and also has the ability to generate diverse captions for comprehensively understanding images.

3.1. Multimodal embedding

To generate captions for an image, we first use a pretrained Faster RCNN [38] model to extract N visual objects and recognise M OCR tokens by the Rosetta OCR [6].

Visual embedding. For the i-th visual object, the Faster RCNN model outputs appearance feature $\mathbf{v}_i^a \in \mathbb{R}^d$ and a 4-dimensional bounding box coordinate $\mathbf{v}_i^b$. To enrich the visual representation, we apply a linear layer f_1 with LayerNorm [3] to project the above features as $\widehat{\mathbf{v}}_i = f_1([\mathbf{v}_i^a, \mathbf{v}_i^b])$, where $[\cdot, \cdot]$ is a concatenation operation.

Token embedding. For each recognised OCR token, we also use its appearance feature $\mathbf{t}_i^a$ and bounding box coordinate $\mathbf{t}_i^b$. Apart from these features, following the M4C-Captioner [40], we adopt two additional text features to further enrich the representations, including FastText feature $\mathbf{t}_i^f$ and PHOC (pyramidal histogram of characters) feature $\mathbf{t}_i^p$. In particular, $\mathbf{t}_i^f$ is a pretrained word-level embedding for written texts while $\mathbf{t}_i^p$ is a character-level embedding for capturing what characters are present in the tokens. Based on the rich representations of OCR tokens, we calculate OCR token features by $\widehat{\mathbf{t}}_i = f_2([\mathbf{t}_i^a, \mathbf{t}_i^b, \mathbf{t}_i^f, \mathbf{t}_i^p])$, where f_2 is a linear layer with LayerNorm to ensure that token embedding has the same scale as visual embedding.

Multimodal embedding fusion. Based on the above, we obtain visual embedding $\widehat{\mathbf{V}}=[\widehat{\mathbf{v}}_1, ..., \widehat{\mathbf{v}}_N]^\top$ and token embedding $\widehat{\mathbf{T}}=[\widehat{\mathbf{t}}_1, ..., \widehat{\mathbf{t}}_M]^\top$. Since both the OCR tokens and visual objects are visual contents and exist in images as a whole, it is necessary to model their interaction. Formally, given $\widehat{\mathbf{V}}$ and $\widehat{\mathbf{T}}$, we use an L_1-layer Transformer module $\Psi(\cdot; \theta_a)$ to obtain more informative features via a self-attention operation as

$$\mathbf{V}, \mathbf{T} = \Psi\left([\widehat{\mathbf{V}}, \widehat{\mathbf{T}}]; \theta_a\right). \tag{1}$$

3.2. Anchor proposal module

Based on the multimodal embeddings $(\mathbf{V}, \mathbf{T})$, existing TextCap methods such as M4C-Captioner [40] simply treat the texts in an image as another kind of visual information and feed them to a captioning module without distinction. However, compared with ambiguous visual information, the texts in images are essential to describe images and thus need to be considered carefully.

To this end, we propose the anchor proposal module (AnPM) to determine which OCR tokens should be paid

more attention to describe. Inspired by the region proposal network (RPN), AnPM first performs anchor prediction among OCR tokens to output a score for each token and choose a series of important tokens as anchors. After that, to model the complex relationship between tokens, AnPM groups relevant tokens to construct the corresponding anchor-centred graph (ACG) for each anchor. Now, we introduce how to construct ACGs in detail.

Anchor prediction. Intuitively, different OCR tokens play different roles in caption generation. However, it is hard to decide which texts should be paid more attention to. To this end, based on the text features $\mathbf{T}$, we apply a linear layer ϕ as an anchor predictor to predict a score for each token as

$$\mathbf{s}_{anchor} = \text{Softmax}\left(\phi(\mathbf{T})\right). \tag{2}$$

In the anchor score $\mathbf{s}_{anchor} \in \mathbb{R}^M$, each element indicates the importance weight of each OCR token. In training, we adopt the OCR token with the highest score as the anchor by argmax operation, denoted as

$$\mathbf{T}_{anchor} = \mathbf{T}_{i,:}, \quad \text{where } i = \text{argmax}(\mathbf{s}_{anchor}). \tag{3}$$

After that, we can obtain the anchor embedding $\mathbf{T}_{anchor}$. During the inference phase, we choose OCR tokens with top-K scores as anchors.

Anchor-centred graph construction. In this paper, we employ a RNN module and take $\mathbf{T}_{anchor}$ as the initial hidden state to model the potential dependence between the anchor and different tokens. The ACG construction for the anchor $\mathbf{T}_{anchor}$ can be formulated as:

$$\begin{aligned} \mathbf{T}_{graph} &= \text{RNN}\left(\mathbf{T},\ \mathbf{T}_{anchor}\right), \\ \mathbf{s}_{graph} &= \sigma(f_3(\mathbf{T}_{graph})), \end{aligned} \tag{4}$$

where $\mathbf{T}_{graph} \in \mathbb{R}^{M\times d}$ denotes the updated token feature and f_3 is a linear layer followed by Sigmoid activation function σ to output the graph score $\mathbf{s}_{graph} \in \mathbb{R}^M$ for all M tokens. After that, we concatenate $\mathbf{T}_{anchor}$ and its relevant tokens to construct the ACG $\mathcal{G}$ as follows:

$$\begin{aligned} \mathcal{G} =& [\mathbf{T}_{anchor}, \{\mathbf{T}^i_{graph}\}], \quad \text{where} \\ & \mathbf{s}^i_{graph} > 0.5,\ \forall i \in [1, M]. \end{aligned} \tag{5}$$

Overall, the anchor proposal module (AnPM) learns to select an important OCR token as an anchor and then construct an ACG for it. In this way, AnPM is able to propose a series of ACGs for an input image, which would be fed into the captioning module as guidance to generate diverse captions. Meanwhile, the generation process of each ACG is independent and will not be affected by other pairs, which greatly improves the quality of the generated captions.

3.3. Anchor captioning module

Compared with general image captioning, the TextCap requires captioning models to not only describe visual objects but also contain OCR tokens in the generated captions. To achieve this, we carefully design a progressive captioning module, namely Anchor Captioning Module (AnCM). Inspired by the Deliberation Network [48], as shown in Figure 2, AnCM consists of a visual-captioner (denoted as AnCM_v) and a text-captioner (denoted as AnCM_t). First, the visual-captioner, a standard image captioning module, uses the updated visual embedding $\mathbf{V}$ to generate a visual-specific caption $\mathcal{Y}'$ with C words, denoted as $\mathcal{Y}' = \{y'_c\}_{c=1}^C$. Then, the text-captioner is proposed to refine the generated caption based on the text information of ACG $\mathcal{G}$ (*see* Eqn. (5)). Following the training of sequence generation task, the goal of AnCM is to maximise the data log likelihood function as follows:

$$\log \sum_{c=1}^{C} P(y_c|\text{AnCM}_t(y'_c, \mathcal{G}))\ P(y'_c|\text{AnCM}_v(\mathbf{V})), \tag{6}$$

where $\{y_c\}$ is the final generated caption. Since the predicted token y'_c is obtained by argmax function which is a non-differentiable operation, we cannot directly optimise the above equation. To address this issue, we feed the hidden state feature $\mathbf{h}_c$ outputted from AnCM_v directly into AnCM_t and the training loss for AnCM_t is computed as follows:

$$\begin{aligned} \mathcal{L}_{tcap} &= -\log \sum_{c=1}^{C} P(y_c|\text{AnCM}_t(\mathbf{h}_c, \mathcal{G}); \theta_t), \quad \text{where} \\ \mathbf{h}_c &= \text{AnCM}_v(\mathbf{V}; \theta_v). \end{aligned} \tag{7}$$

The θ_v and θ_t are learnable parameters of visual-captioner and text-captioner, respectively. In this way, we can train AnCM in an end-to-end manner. Next, we will introduce more details about the two captioners.

Visual-captioner (AnCM_v). To capture long-range dependency in sequence modelling, we employ an L_2-layer Transformer module (Ψ) as the backbone of the visual-captioner. Specifically, the visual-captioner generates tokens in an auto-regressive manner as follows:

$$\begin{aligned} \mathbf{h}_c &= \Psi(\mathbf{V}, \text{LM}(\mathbf{y}'_{c-1}); \theta_v), \\ y'_c &= \text{argmax}(f_4(\mathbf{h}_c)), \end{aligned} \tag{8}$$

where $\mathbf{y}'_{c-1}$ denotes the embedding of previous output token, $f_4(\cdot)$ is a linear classifier for common vocabulary and we can obtain the predicted token y'_c with argmax operation. Here, we use the prefix language modelling (LM) technique [37] to ensure that the input entries only use previous predictions, and avoid peeping at subsequent generation processes. Thus far, with the help of the visual-captioner,

we obtain a visual-specific caption $\{y'_c\}$ and its hidden state feature $\{\mathbf{h}_c\}_{c=1}^{C}$. Formally, we define the training loss for AnCM_v as $\mathcal{L}_{vcap} = -\log \sum_{c=1}^{C} P(y'_c)$.

Image captioning is a fairly mature sequence-generation task, and researchers have also proposed many models with promising performance. In this work, we do not focus on designing a new captioning module. Intuitively, the backbone of visual-captioner can be easily replaced by other image captioning models, such as BUTD [2] and AoANet [21].

Text-captioner (AnCM_t). Based on the hidden state features $\{\mathbf{h}_c\}_{c=1}^{C}$, the text-captioner aims to generate text-specific captions that contain given OCR tokens. To this end, at this stage, we use ACG as the guidance to refine the caption generated by the visual-captioner from the last step. Specifically, we use an L_3-layer Transformer module (Ψ) as the backbone of the text-captioner. Relying on self-attention, the Transformer allows multimodal embedding to freely interact with others, thereby achieving satisfactory progress in sequence generation tasks. Formally, given the hidden state and ACGs, our AnCM_t can output the joint embedding as follows:

$$\widehat{\mathcal{G}},\ \widehat{\mathbf{y}}_c = \Psi([\mathcal{G}, \mathbf{h}_c, \text{LM}(\mathbf{y}_{c-1})]; \theta_t), \tag{9}$$

where $\widehat{\mathcal{G}}$ is the updated token embedding in the ACG and $\widehat{\mathbf{y}}_c$ is the embedding of c-th prediction. Following M4C [19], we adopt different classifiers for common vocabulary and OCR candidate tokens as

$$y_c = \text{argmax}([f_4(\widehat{\mathbf{y}}_c),\ f_{dp}(\widehat{\mathcal{G}}, \widehat{\mathbf{y}}_c)]), \tag{10}$$

where f_4 is the shared classifier with visual-captioner in Eqn. (8), f_{dp} denotes the dynamic pointer network [19] that makes prediction based on the $\widehat{\mathcal{G}}$ and $\widehat{\mathbf{y}}_c$. After concatenating two prediction scores, we use the argmax function on the final prediction score to obtain the predicted token y_c.

Compared with general captioning methods, the proposed AnCM makes full use of a key character of the TextCap task, that is, the OCR token can be used as an important guide to improve the generation accuracy.

3.4. Training details

Formally, we train our Anchor-Captioner model by optimising the following loss function:

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{anchor}(\mathbf{s}_{anchor}) + \alpha\, \mathcal{L}_{graph}(\mathbf{s}_{graph}) \\ +\ \beta\, \mathcal{L}_{vcap}(\mathcal{Y}') + \eta\, \mathcal{L}_{tcap}(\mathcal{Y}),\end{aligned} \tag{11}$$

where the $\mathbf{s}_{anchor/graph}$ is the output of AnPM (*see* Eqn. (2) or (4)), the $\mathcal{Y}'=\{y'_c\}$ and $\mathcal{Y}=\{y_c\}$ denote the generated visual-specific caption and text-specific caption (*see* Eqns. (8) and (10)), respectively. $\{\alpha, \beta, \gamma\}$ are trade-off parameters. In practice, all the above four losses adopt the binary cross-entropy loss function. We train AnPM with $\mathcal{L}_{anchor}$ and $\mathcal{L}_{graph}$ to find the most frequently described ACGs. We train AnCM with $\mathcal{L}_{vcap}$ and $\mathcal{L}_{tcap}$ to generate visual-specific and text-specific captions. Due to the page limitation, we put detailed training and inference algorithms in Supplementary A.

Ground-truth labels. Next, we will illustrate how to obtain supervisions for training AnPM and AnCM. 1) Given a manually annotated caption (e.g.,'*a man ... number 15 on it*' in Figure 2), we consider it as ground truth (gt) for $\mathcal{L}_{tcap}$ to train the text-captioner. 2) We then mask the OCR tokens contained in the manually generated caption with [unk] to build a new caption (e.g.,'*a man ... number [unk] on it*') for $\mathcal{L}_{vcap}$ to train the visual-captioner. This is because we do not require the AnCM_v to make prediction among OCR tokens by using only the visual information. 3) Considering that different people will describe the same image from different views, the most frequently described token is often the most important one for describing an image. Thus, we choose the most frequently described token as gt for $\mathcal{L}_{anchor}$ to train the anchor prediction module in AnPM. 4) Given the chosen anchor, we consider the tokens that appear in the same caption as gt for $\mathcal{L}_{graph}$ to train AnPM to find the most relevant tokens for an anchor. During the test phase, we do not need to construct gt-ACGs since they are only used for calculating losses in the training. Note that the gt-ACGs are automatically mined and constructed from the same training split without using any additional annotations, and thus comparisons with other methods are fair.

4. Experiments

We verify the effectiveness of our method on the TextCaps [40] dataset. In the following, we first briefly introduce the TextCaps and the comparison settings for it in Sec. 4.1. More implementation details can be found in the Sec. 4.2. And then, we compare our method with existing captioning models in Sec. 4.3 and Sec. 4.4. Last, we demonstrate our proposed method by providing some visualisation results and analysis in Sec. 4.5.

4.1. Datasets and settings

Datasets. The TextCaps dataset [40] is collected from Open Image V3 dataset and contains 142,040 captions on 28,408 images, which have been verified to contain text through the Rosetta OCR system [6] and human annotators. For each image, there are five independent captions. In the test split, each image has an additional caption that is collected to estimate human performance on the dataset. The dataset also contains captions where OCR tokens are not presented directly but are used to infer a description [40]. In this case, the captioning models are required to perform challenging reasoning rather than simply copy the OCR tokens. Most

#	Method	TextCaps validation set metrics				
		B	M	R	S	C
1	BUTD	20.1	17.8	42.9	11.7	41.9
2	AoANet	20.4	18.9	42.9	13.2	42.7
3	M4C-Captioner	23.3	22.0	46.2	15.6	89.6
4	M4C-Captioner$^-$	15.9	18	39.6	12.1	35.1
5	AnCM$_v$	16.1	16.3	40.1	11.2	29.1
6	**Ours**	**24.7**	**22.5**	**47.1**	**15.9**	**95.5**
#	Method	TextCaps test set metrics				
		B	M	R	S	C
7	BUTD	14.9	15.2	39.9	8.8	33.8
8	AoANet	15.9	16.6	40.4	10.5	34.6
9	M4C-Captioner	18.9	19.8	43.2	12.8	81.0
10	MMA-SR	19.8	20.6	44.0	13.2	**88.0**
11	**Ours**	**20.7**	**20.7**	**44.6**	**13.4**	87.4
12	Human	24.4	26.1	47.0	18.8	125.5

Table 1. Comparison with SOTA methods on the validation and test set. In particular, rows 4 are captioning models without OCRs, i.e., only use visual information to generate captions. The last row is the estimated human performance, which can be seen as the upper bound of captioning models on the TextCaps dataset.

captions contain two or more OCR tokens, and the average length of captions is 12.4.

Evaluation metrics. We use five standard evaluation metrics in image captioning, including BLEU (B) [36], METEOR (M) [10], ROUGE_L (R) [27], SPICE (S) [1] and CIDEr (C) [43] to evaluate the accuracy. Following the benchmark setting [40], we mainly focus on CIDEr, which puts more weight on informative tokens and is more suitable for this dataset. To evaluate the diversity of generated captions, we use Div-n [26] and SelfCIDEr [46] metrics on the validation set. In particular, the Div-n focuses on token-level diversity while the SelfCIDEr is used for semantic-level diversity. In addition, we propose a new metric, called Cover Ratio (CR), to measure the content diversity, that is, how many OCR tokens are included in the generated captions. For notation convenience, we omit the percentage in the metric scores.

Compared methods. We first compare our method with two state-of-the-art (SOTA) image captioning methods, i.e., BUTD [2] and AoANet [21]. For the TextCap task, we compare our method with the current SOTA methods M4C-Captioner [40] and MMR-SA [45]. For fair comparisons, we use the same dataset annotation and multimodal feature extraction methods (including the OCR system and Faster RCNN) for all considered methods in our experiments. We conduct extensive experiments on the validation and test set. In particular, the evaluation results are provided by the test server of the TextCaps-Challenge[1]. Since the number of submissions of the results on the test set is limited, we conduct ablation studies on the validation set.

[1]TextCaps: https://textvqa.org/textcaps

#	Method	Div-1	Div-2	selfCIDEr	CR
1	BUTD	27.0	36.4	45.3	-
2	M4C-Captioner	27.2	41.2	49.4	27.3
3	**Ours**	**29.8**	**43.8**	**57.6**	**37.8**
4	Human	62.1	87.0	90.9	19.3

Table 2. Diversity analysis. The BUTD and M4C-Captioner generate diverse captions via beam search (beam size is 5).

#	Projection	B	M	R	S	C	A	F1
1	Single	23.9	22.2	46.7	15.6	90.3	48.4	68.8
2	Multiple	23.7	22.4	46.3	16.0	90.7	49.0	68.9
3	**Sequence**	**24.7**	**22.5**	**47.1**	**15.9**	**95.5**	**49.1**	**71.8**

Table 3. Ablation studies of anchor proposal module (AnPM) with independent projection (FC) and sequence projection (RNN). Apart from using captioning metrics, we also use accuracy (A) and F1 score to further measure the performance of AnPM.

4.2. Implementation details

In our implementation[2], the feature dimension d is 768 and the f_* is a linear layer with LayerNorm activation function to stabilise training. We train our model for 12,000 iterations with a batch size of 128. During training, we use the Adamax optimiser [23] with a learning rate of 2e-4. We adopt default parameter settings of BERT-BASE [12] for the transformer module Ψ, such as 12 self-attention heads. But the number of stacked layers are $L_1 = 2, L_2 = L_3 = 4$, respectively. For fair comparisons, following the TextCaps benchmark [40], we use the same fixed common vocabulary and the feature embeddings of visual objects and OCR tokens. The number of visual objects is $N = 100$ and the number of OCR tokens is $M = 50$. The maximum generation length is $C = 30$. The trade-off parameters of different losses are set to $\alpha = \beta = \gamma = 1$.

4.3. Main results

Overall results. As shown in Table 1, we first compare our method with the current SOTA captioning models, including BUTD, AoANet, and M4C-Captioner. From the table, the BUTD and AoANet, standard image captioning models, show poor performances on the validation set since they fail to describe the texts in images. M4C-Captioner reasons over multimodal information and outperforms standard image captioning models by a large margin. Compared with M4C-Captioner, our model improves the CIDEr score from 89.6 to 95.5 on the validation set and achieves 6 absolute improvement on the test set. In particular, we also report the result of visual-captioner AnCM$_v$ (row 5), which can be seen as a degraded version of our model without using OCR tokens. Same as M4C-Captioner w/o OCRs (row 4), AnCM$_v$ is hard to generate reliable captions for the TextCaps dataset. To address this issue, our model is equipped with an additional text-captioner that refines gen-

[2]https://github.com/guanghuixu/AnchorCaptioner.

#	Anchor	ACG	B	M	R	S	C
1		All	21.2	21.0	44.8	14.5	76.6
2	Large	Around	21.4	21.1	44.9	14.4	77.4
3		Random	20.8	20.7	44.4	14.1	72.6
4		All	21.2	21.0	44.8	14.5	76.6
5	Centre	Around	21.5	21.2	45.0	14.4	78.0
6		Random	20.7	20.8	44.5	14.1	73.1
7	-	All	21.1	21.1	44.7	14.6	76.7
8		Random	20.4	20.6	44.1	13.9	70.2
9		All	23.5	22.4	46.3	15.7	90.3
10	GT	Around	22.1	21.9	45.6	15.2	83.9
11		Random	21.4	21.2	45.0	14.7	78.7
12	AnPM	AnPM	24.7	22.5	47.1	15.9	95.5
13	**GT**	**GT**	**25.6**	**23.4**	**48.1**	**16.9**	**104.9**

Table 4. Ablation studies of ACG construction using rule-based approaches. For instance, row 2 ('Large'+'Around') means that we choose an OCR token with the largest region size as the anchor, and then group the five closest tokens to construct its ACG. In particular, we randomly group some tokens into an ACG, denoted as 'Random' in the table. 'GT' denotes ground-truth and 'AnPM' means using the prediction of AnPM.

erated captions with the text information. For fair comparisons, we choose the ACG with the highest anchor score to refine the generated caption in this experiment, since existing methods derive only one caption for each input image. In this way, our full model further boosts the CIDEr score from 29.1 to 95.5 in the validation set.

Diversity analysis. To further evaluate the diversity of the generated captions, we compare Anchor-Captioner with BUTD and M4C-Captioner in terms of diversity metrics. Since existing methods only generate a global description for an image, we use the beam search technique for them to produce diverse captions as baselines, where the beam size is set to 5. For fair comparisons, in our method, we also sample five ACGs for each image to generate captions. As shown in Table 2, our method surpasses baselines in terms of all considered metrics. Interestingly, the ground-truth captions (by humans) have high selfCIDEr but with low OCR cover ratio (CR). It means that humans may tend to describe the salient image contents but ignore some OCR tokens. Compared with human captioning, our method is able to generate multiple captions with content diversity, covering more OCR tokens. Note that, cover ratio (CR) score for BUTD method is empty, because OCR tools are not used in it.

4.4. Ablation studies

In this section, we further conduct ablation studies to demonstrate the effectiveness of AnPM and AnCM.

For AnPM, we first compare three different kinds of ACG construction strategies, i.e., independent projection (FC), multiple projection (transformer module) and sequence projection (RNN module). As shown in Table 3, the **RNN** outperforms **FC** in terms of all considered metrics, especially improves the CIDEr score from 90.3 to 95.5. As discussed in Sec. 3.2, the sequence projection is more reasonable since it considers the history prediction. More details can be found in the supplement material. Moreover, we also report the accuracy of anchor prediction and the F1 score of the predicted ACG. Note that, there is a trade-off between obtaining high F1 score and diversity. To achieve high accuracy and F1 score, AnPM tends to predict the most frequently described ACG, which, however, could suffer from low diversity of generated captions.

#	Method	B	M	R	S	C
1	M4C-Captioner	23.3	22.0	46.2	15.6	89.6
2	M4C-Captioner†	24.1	22.6	46.7	15.7	93.8
3	M4C-Captioner*	24.4	22.6	46.9	15.8	99.6
4	AnCM$_v$ + AnCM$_t^\dagger$	24.7	22.5	47.1	15.9	95.5
5	AnCM$_v$ + AnCM$_t^*$	**25.6**	**23.4**	**48.1**	**16.9**	**104.9**

Table 5. Ablation studies of Anchor Caption Module (AnCM). $\dagger$ denotes captioning modules using prediction ACGs provided by AnPM, while $*$ denotes captioning modules using ground-truth.

In addition to the above comparisons, we also compare AnPM (RNN projection) with the rule-based ACG construction and report the quantitative results in Table 4. To be specific, we first adopt different rules to select token as an anchor, including the largest token (rows 1-3), the centre token (rows 4-6), the ground-truth anchor (rows 9-11). Then, we choose tokens to construct ACG using different strategies (i.e., 'All / Around / Random'). In particular, we try to group tokens into a graph directly without performing anchor selection process (in rows 7-8). From the table, all the rule-based methods suffer low metric performance even given the GT anchor to construct ACGs. The learning-based method (AnPM) outperforms rule-based methods by a large margin. One reason is that our AnPM considers the rich semantic information of the tokens themselves and the visual information in images, while the rule-based approaches mainly use shallow information such as size and location.

We also conduct ablation experiments for AnCM. From the results in Table 5, we draw the following main observations. 1) As shown in rows 1-3, the M4C-Captioner† and M4C-Captioner* that take the predicted ACGs and ground-truth ACGs as inputs, outperform the original M4C-Captioner by around 4 and 10 in terms of the CIDEr score, respectively. These results well verify our idea, i.e., first group OCR tokens into different ACG and then describe each ACG with a specific caption. 2) Compared with M4C-Captioner (row 1), our method improves CIDEr score from 89.6 to 95.5. 3) Equipped with AnPM, the M4C-Captioner† (row 2) achieves better performance, which implies that our AnPM can be easily extended to existing text-based reasoning methods. 4) Even for the same ACG inputs, our method is still superior to M4C-Captioner† and M4C-Captioner*, which demonstrates the powerful captioning ability of our

M4C: a person is holding a green laptop with a green screen that says blobo
AnCM$_v$: a person is holding a compu-ter computer monitor
Ours-1: a person is holding a blobo game on a computer screen
Ours-2: a blobo game is being held in front of a computer screen
Ours-3: a computer screen shows a game called blobo

M4C: a man stands at a podium at a podium that says firefox
AnCM$_v$: a man is standing a presentation on a screen that says
Ours-1: a man is giving a presentation with a screen that says mozilla
Ours-2: a man is giving a presentation with a screen that says " earn & keep "
Ours-3: a man is giving a presentation with a screen that says " earn your keep trust "

M4C: a poster for the mayan starts and theatre
AnCM$_v$: a poster for a <unk> <unk> shows a man of a man in the top
Ours-1: a poster for mayan theatre at the top of the page
Ours-2: a poster for mayan hill and 11th street
Ours-3: a poster for mayan and theatre shows a picture of a man on the bottom

M4C: a phone with the word lg on the screen
AnCM$_v$: a orange phone with a red of a phone and a sign that says it
Ours-1: an orange lg phone with a screen that says ' lg ' on it
Ours-2: an orange phone with the word jazz on it
Ours-3: a phone screen shows a time of 567 : 00 on it

Figure 3. Visualisation results on the TextCaps validation set. The prediction results of M4C-Captioner (M4C), visual-captioner (AnCM$_v$) and the proposed Anchor-Captioner are placed below the images in turn. The <unk> denotes 'unkown' token. For better visualisation, the underlined word is copy from OCR tokens. In particular, Anchor-Captioner will refine the caption generated by AnCM$_v$. The modified tokens are viewed in red colour.

AnCM. 5) According to the last two rows, our AnCM suffers a performance degradation with the predicted ACGs as input, indicating that our method still has great potentials for improving.

4.5. Visualisation analysis

To further demonstrate the effectiveness of our method, we show some visualisation results on the TextCaps validation set. From Figure 3, our Anchor-Captioner is able to refine the rough captions generated by the visual-captioner (AnCM$_v$). Specifically, for each input image, AnCM$_v$ first uses visual information to generate a global caption, such as 'a man' and 'a poster'. Similar to general image captioning models, AnCM$_v$ is difficult to describe the texts in images. As a result, we can see the visual-specific captions may contain some <unk> tokens. It means that AnCM$_v$ cannot use limited information to generate reasonable predictions in this case. And then, Anchor-Captioner use anchor-centred graphs (ACGs) to further refine the visual-specific captions. Note that, the refine process is not only to replace the <unk> token, but also to revise the entire caption. There are 66.39% of generated captions with <unk>, and each caption has 1.24 <unk> on average. AnCM$_t$ modified 26.85% of words on the AnCM$_v$'s output and improved CIDEr score from 29.1 to 95.5 (*see* Tabel 1). We also randomly sample different ACGs to demonstrate the diversity of our generation. Compared with M4C-Captioner, our method is able to generate fine-grained captions and cover more OCR tokens. To further demonstrate the controllability and diversity of our method, we provide more visualisation results in the supplement material.

5. Conclusion

In this paper, we have proposed an Anchor-Captioner to resolve the TextCap task. To solve this task, existing methods tend to generate only one rough global caption which contains one or two salient objects in the complex scene. Intuitively, such methods may ignore some regions that people are really interested in. Unlike existing methods, we seek to generate multiple captions from different views and cover more valuable scene information. Specifically, we first propose an anchor proposal module to group OCR tokens and construct anchor-centred graphs (ACGs) by modelling the relationship between image contents. After that, our anchor captioning module first generates a rough visual-specific caption and then uses the above ACGs to further refine it to multiple text-specific captions. In this way, our method is able to generate diverse captions to cover more information in images. Our method achieves state-of-the-art performance on the TextCaps dataset and outperforms the benchmark by 6 in terms of CIDEr score. Extensive ablation experiments also verify the effectiveness of each component of our method. Note that our anchor captioning module has the potential to solve both image captioning and text-based image captioning tasks simultaneously, which we leave to our future study.

Acknowledgements. This work was partially supported by the Science and Technology Program of Guangzhou, China, under Grant 202007030007, Fundamental Research Funds for the Central Universities D2191240, Program for Guangdong Introducing Innovative and Enterpreneurial Teams 2017ZT07X183, Opening Project of Guangdong Key Laboratory of Big Data Analysis and Processing.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. Spice: Semantic propositional image caption evaluation. In *ECCV*, pages 382–398, 2016.

[2] Peter Anderson, X. He, C. Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, pages 6077–6086, 2018.

[3] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. In *CoRR*, 2016.

[4] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *CVPR*, pages 9365–9374, 2019.

[5] Ali Furkan Biten, Ruben Tito, Andrés Mafla, Lluís Gómez, M. Rusiñol, Ernest Valveny, C. Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *ICCV*, pages 4290–4300, 2019.

[6] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *ACM SIGKDD*, pages 71–79. ACM, 2018.

[7] Shizhe Chen, Qin Jin, Peng Wang, and Qi Wu. Say as you wish: Fine-grained control of image caption generation with abstract scene graphs. In *CVPR*, pages 9959–9968, 2020.

[8] Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Show, control and tell: A framework for generating controllable and grounded captions. In *CVPR*, pages 8307–8316, 2019.

[9] Chaorui Deng, Ning Ding, Mingkui Tan, and Qi Wu. Length-controllable image captioning. In *ECCV*, volume abs/2007.09580, 2020.

[10] Michael J. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *WMT-ACL*, pages 376–380, 2014.

[11] Aditya Deshpande, Jyoti Aneja, Liwei Wang, Alexander G. Schwing, and David A. Forsyth. Fast, diverse and accurate image captioning guided by part-of-speech. In *CVPR*, pages 10695–10704, 2019.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019.

[13] I. Fine. Sensory systems: Do you hear what i see? *Nature*, 508:461–462, 2014.

[14] Chuang Gan, Zhe Gan, X. He, Jianfeng Gao, and L. Deng. Stylenet: Generating attractive visual captions with styles. In *CVPR*, pages 955–964, 2017.

[15] Zhe Gan, Chuang Gan, X. He, Y. Pu, K. Tran, Jianfeng Gao, L. Carin, and L. Deng. Semantic compositional networks for visual captioning. In *CVPR*, pages 1141–1150, 2017.

[16] Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. In *EMNLP-IJCNLP*, pages 6111–6120, 2019.

[17] Longteng Guo, J. Liu, Peng Yao, Jiangwei Li, and H. Lu. Mscap: Multi-style image captioning with unpaired stylized text. In *CVPR*, pages 4204–4213, 2019.

[18] Yong Guo, Yin Zheng, Mingkui Tan, Qi Chen, Jian Chen, Peilin Zhao, and Junzhou Huang. Nat: Neural architecture transformer for accurate and compact architectures. In *Advances in Neural Information Processing Systems*, pages 735–747, 2019.

[19] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *CVPR*, pages 9992–10002, 2020.

[20] Deng Huang, Peihao Chen, Runhao Zeng, Qing Du, Mingkui Tan, and Chuang Gan. Location-aware graph convolutional networks for video question answering. In *The AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[21] Lun Huang, Wenmin Wang, J. Chen, and Xiao-Yong Wei. Attention on attention for image captioning. In *ICCV*, pages 4633–4642, 2019.

[22] Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, pages 4565–4574, 2016.

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, pages 4190–4198, 2014.

[24] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123:32–73, 2016.

[25] Jason Lee, Elman Mansimov, and Kyunghyun Cho. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *EMNLP*, volume abs/1802.06901, pages 1173–1182, 2018.

[26] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *HLT-NAACL*, pages 110–119, 2016.

[27] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *ACL*, pages 74–81, 2004.

[28] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, pages 740–755, 2014.

[29] L. Liu, Mengge He, G. Xu, Mingkui Tan, and Qi Wu. How to train your agent to read and write. *ArXiv*, abs/2101.00916, 2021.

[30] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Improved image captioning via policy gradient optimization of spider. In *ICCV*, pages 873–881, 2017.

[31] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *CVPR*, pages 9806–9815, 2020.

[32] Jiasen Lu, Jianwei Yang, Dhruv Batra, and D. Parikh. Neural baby talk. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7219–7228, 2018.

[33] Alexander Patrick Mathews, Lexing Xie, and Xuming He. Semstyle: Learning to generate stylised image captions using unaligned text. In *CVPR*, pages 8591–8600, 2018.

[34] A. Mishra, Shashank Shekhar, A. Singh, and A. Chakraborty. Ocr-vqa: Visual question answering by reading text in images. *ICDAR*, pages 947–952, 2019.

[35] Shuaicheng Niu, J. Wu, Yi-Fan Zhang, Yong Guo, P. Zhao, Junzhou Huang, and Mingkui Tan. Disturbance-immune weight sharing for neural architecture search. *ArXiv*, abs/2003.13089, 2020.

[36] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *ACL*, page 311–318, 2002.

[37] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*, 21(140):1–67, 2020.

[38] Shaoqing Ren, Kaiming He, Ross B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *TPAMI*, 39:1137–1149, 2017.

[39] Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *CVPR*, pages 1179–1195, 2017.

[40] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: A dataset for image captioning with reading comprehension. In *ECCV*, pages 742–758, 2020.

[41] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *CVPR*, pages 8317–8326, 2019.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[43] Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, pages 4566–4575, 2015.

[44] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, pages 3156–3164, 2015.

[45] Jing Wang, Jinhui Tang, and Jiebo Luo. Multimodal attention with image text spatial relationship for ocr-based image captioning. In *ACM MM*, page 4337–4345, 2020.

[46] Qingzhong Wang and Antoni B. Chan. Describing like humans: On diversity in image captioning. In *CVPR*, pages 4195–4203, 2019.

[47] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[48] Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. Deliberation networks: Sequence generation beyond one-pass decoding. In *NeurIPS*, pages 1784–1794, 2017.

[49] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C. Courville, R. Salakhutdinov, R. Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, volume abs/1502.03044, pages 2048–2057, 2015.

[50] Runhao Zeng, Haoming Xu, Wenbing Huang, Peihao Chen, Mingkui Tan, and Chuang Gan. Dense regression network for video grounding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.