How Far Can We Go with Scene Descriptions for Visual Question Answering?

Yusuke Hirota[1,a] Noa Garcia[1,b] Mayu Otani[3,c] Chenhui Chu[2,d] Yuta Nakashima[1,e]
Ittetsu Taniguchi[1,f] Takao Onoye[1,g]

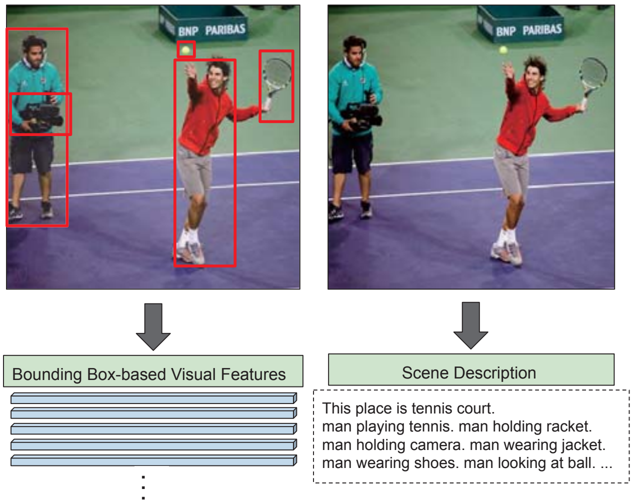
Abstract:
Visual question answering (VQA) is the task of answering questions about an image's visual content. To represent images, the bounding box-based visual representations have been widely used as the de-facto standard. In contrast, the recent progress in Transformer language models has made it possible to represent simultaneously inter-relationships between two consecutive sentences as well as intra-relationships between the individual words in a sentence. The outstanding performance of such language models in multiple language-based tasks inspired us to consider textual representations of images for VQA. Thus, instead of using visual features directly extracted from images, we propose to generate scene descriptions by using state-of-the-art recognition models. Results on VQA-CP v2 show our proposed textual descriptions have the potential to be a faithful representation for VQA. Even so, our experiments reveal there is still room for improvement in our generated scene descriptions.

Keywords: visual question answering, scene descriptions, Transformer

1. Introduction

Vision and language is a research area that has been increasingly drawing more attention. One of the main tasks of vision and language is visual question answering (VQA) [1, 2]. VQA aims to answer questions about an image's visual content, requiring a computer system to understand both a question and an image. For understanding the visual content, the bounding box (region)-based visual representation has been used as the de-facto standard due to bottom-up attention's success [3]. However, by handling the visual features extracted from bounding boxes independently, a machine often fails to describe the visual relationships, or, even if the prediction is correct, he regions that should be paid attention to answer the question are not appropriate [4].

On the other hand, the recent progress in Transformer language models [5,6] outperform previous techniques on various key NLP datasets such as GLUE [7] and SQuAD [8]. For this reason, textual representation of images might be a powerful tool when understanding visual contents, providing a deep understanding of, e.g., interactions among objects in images. Image captioning [9], which is the process of generating textual description according to the content observed in an image, is one of the major tasks to generate textual representation of an image. However, currently, most work on VQA is focused on leveraging bounding box-based visual representation, and application of textual representations of images to VQA is underexplored.



This place is tennis court.
man playing tennis. man holding racket.
man holding camera. man wearing jacket.
man wearing shoes. man looking at ball. ...

Fig. 1: Typically, VQA models use deep visual feature vectors extracted based on object detectors as the representation of an image (left). Our model generates a scene description instead of the visual feature vectors. (right)

In this work, we delve into the effectiveness of textual representation of images. Specifically, as shown in **Fig. 1**, we explore to generate scene descriptions as the representation of an image by leveraging state-of-the-art image recognition techniques. The generated scene descriptions and a question are jointly fed into a

[1] Osaka University
[2] Kyoto University
[3] CyberAgent, Inc.
[a] y-hirota@ist.osaka-u.ac.jp
[b] noagarcia@ids.osaka-u.ac.jp
[c] otani_mayu@cyberagent.co.jp
[d] chu@i.kyoto-u.ac.jp
[e] n-yuta@ids.osaka-u.ac.jp
[f] i-tanigu@ist.osaka-u.ac.jp
[g] onoye@ist.osaka-u.ac.jp

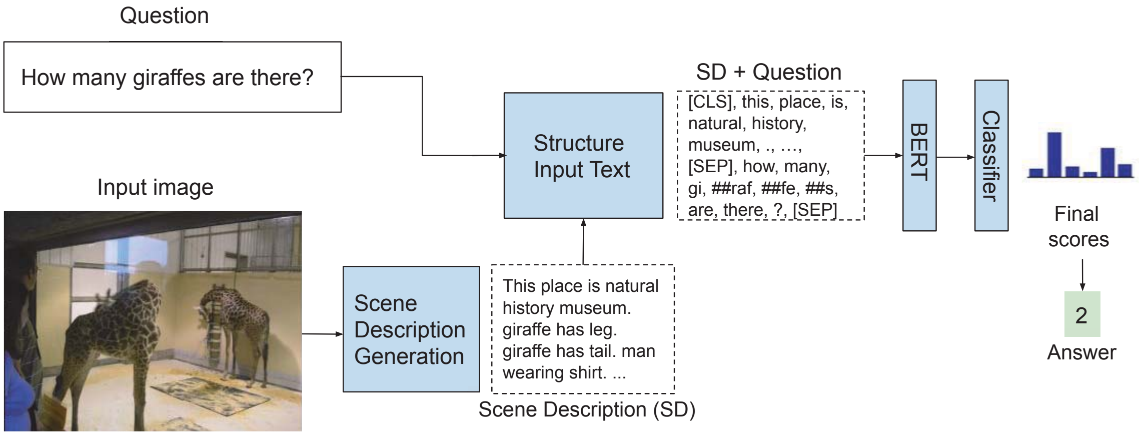

Question
How many giraffes are there?
Input image
Scene Description Generation
This place is natural history museum. giraffe has leg. giraffe has tail. man wearing shirt. ...
Scene Description (SD)
Structure Input Text
SD + Question
[CLS], this, place, is, natural, history, museum, ., …, [SEP], how, many, gi, ##raf, ##fe, ##s, are, there, ?, [SEP]
BERT
Classifier
Final scores
2
Answer

Fig. 2: Model overview.

pre-trained Transformer language model to answer a given question. We extensively evaluate our model on a well-known dataset, VQA-CP v2 [10]. Additionally, we analyze our experimental results in detail. For the current progress, our scene description is not competitive with the existing deep visual features. Still, we found that textual descriptions have potential benefits that can represent the relationships among objects more directly than previous visual features.

2. Related Work

We develop a model for VQA that takes advantage of image content by generating scene descriptions. In what follows, we first review work on image representations for various vision and language tasks, such as VQA, and we introduce Transformers.

2.1 Image Representations

Deep Visual Features In vision and language tasks, visual features have played a key role in leveraging the visual content of images. Most VQA methods use deep visual features extracted by vision models pre-trained on ImageNet [11] and Visual Genome (VG) [12]. There are roughly two types of deep visual features obtained by vision models: 1) grid-based visual features and 2) bounding box (region)-based visual features. Grid-based visual features are convolutional feature maps typically from VGG [13] or ResNet [14]. Grid-based visual features are vectors, each of which represents a uniform grid region in the image, without being aware of its content. On the other hand, bounding box-based visual features, which is extracted by object detectors, such as Faster R-CNN [15], are a set of prominent image regions, with each region represented by a pooled convolutional feature vector. Bounding box-based visual features possibly allow to pay attention to the critical parts in the image.

Cross-Modal Representation Building on top of the recent progress in language models, some work has adapted Transformer language models to fuse visual and textual information for vision and language tasks. Recent studies [16–20] pre-train multi-layer Transformers by concatenating bounding box-based visual features and text features as input. These models can actually learn general cross-modal representations and result in state-of-the-art results in downstream vision and language tasks.

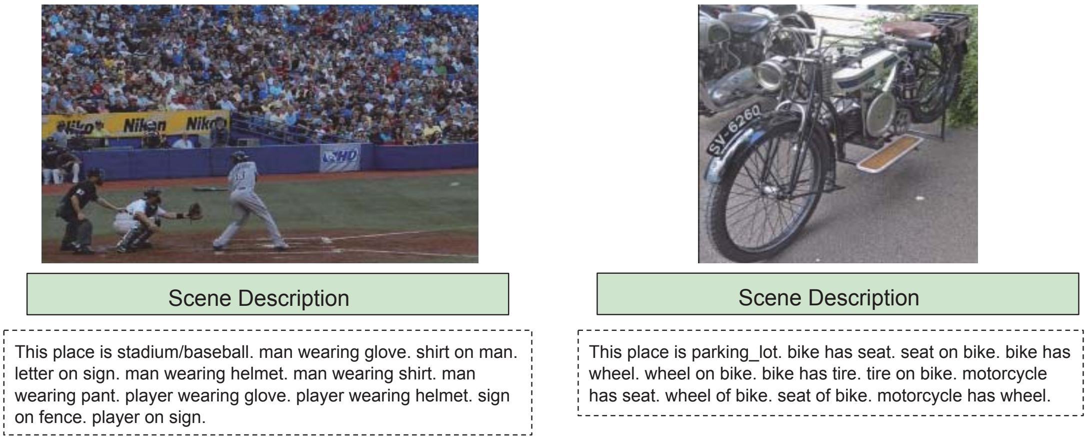
Textual Representation There are some methods that use textual representation of images or videos for VQA. They can succinctly encode the necessary information to answer the questions. Garcia et al. [21] generates video scene descriptions in an unsupervised manner by first generating scene graphs to represent the video scene's content. The generated video scene descriptions are then fed into a Transformer to make a prediction. Sariyildiz et al. [22] proposes a proxy task to learn visual representations from scratch given image-caption pairs, based on the observation that captioned images are easily crawlable. As a result, visual representations that can be transferred well to various downstream tasks. Wu et al. [23] generates image descriptions that contain information directly relevant to a particular VQA question, and the descriptions are exploited to help answer a specific visual question. Ramachandran et al. [24] answers questions by first translating an image to natural language text-based on dense captioning and then answering the question based on the text. The use of textual representation can improve accuracy in the sense that it can easily include attributes and relationships of multiple objects in sentences.

However, almost all of VQA methods described above leverage deep visual features, while few rely only on textual representation for the representation of images. In this paper, we only use textual representations for the visual features of images. We rely on object relationships in images to generate scene descriptions.

2.2 Transformers

Transformers [25] are extensively used for language modeling and entirely rely on self-attention mechanisms to compute representations of their input and output without using sequence-aligned RNNs or convolution. Additionally, pre-trained Transformers can be fine-tuned easily, achieving state-of-the-art performance in many natural language processing tasks [5, 6].

Given the input sequence of word tokens, the input sequence is processed before entering the model by 1) inserting a [CLS] token at the beginning of the first sentence and [SEP] tokens at the



This place is stadium/baseball. man wearing glove. shirt on man. letter on sign. man wearing helmet. man wearing shirt. man wearing pant. player wearing glove. player wearing helmet. sign on fence. player on sign.

This place is parking_lot. bike has seat. seat on bike. bike has wheel. wheel on bike. bike has tire. tire on bike. motorcycle has seat. wheel of bike. seat of bike. motorcycle has wheel.

Fig. 3: Example of generated scene description.

end of each sentence, 2) adding segment embeddings and position embeddings. [CLS] is the classification token used to obtain the output representation. [SEP] is the separator token for separating sentences. The processed input sequence is transformed into learned vector representation of each word and added segment embeddings and position embeddings to get an input embedding. In each self-attention layers of a Transformer, the input embedding is encoded into a representation that holds the learned information for that entire sequence.

3. Our Method

The structure of our model, which aims at answering questions by leveraging scene descriptions, is shown in **Fig. 2**. Given an image and question as input, we first generate a scene description of the image [21]. The scene description and the question are encoded through a Transformer with several self-attention layers. Then, the output from the Transformer is fed into a classifier.

3.1 Scene Description Generation

We use textual representation for the representation of images. We propose a way to generate scene descriptions of images built on top of state-of-the-art image recognition techniques. There are two techniques we use for scene descriptions: place classification and object relationship detection. We generate scene descriptions from the results of these techniques. Below, we first explain the image recognition techniques and describe the detail of the scene descriptions generation process.

Place Classification Place classification is a task to identify a location in an image. As the model to detect where the scene in an image is located, we use the pre-trained Place365 [28] network with ResNet50 [14] backbone. There are 365 scene categories, and we use the network's output as the predicted place.

Object Relationship Detection Object relationship detection is a task to detect the objects that occur in an image and their relations. We use the large-scale visual relationship understanding (VRU) [29] pre-trained on the VG200 dataset [30], a subset of VG with 150 object and 50 relation categories. VRU takes an image as input and predicts multiple relationships and localizes the objects in the image, producing a list of subject-relation-object triplets, bounding boxes, and a prediction score for each triplet.

Scene Description Generation Scene descriptions are generated from the output of place classification and object relationship detection. We obtain the predicted place, P, from pretrained Place365, and a list of N subject-relation-object triplets, $T = \{T_i \mid i = 1, ..., N\}$ with $T_i = (S_i,\ R_i,\ O_i)$ and S_i, R_i, and O_i denote subject, relation, and object respectively from object relationship detection. We generate a sentence, "This place is P." for a place description. We also generate sentences, "$S_i\ R_i\ O_i$." from the results of the triplets. Specifically, we generate sentences "$S_i\ R_i\ O_i$." by removing the duplicate triplets. The maximum number of the sentences is set to ten. We concatenate all sentences to generate scene descriptions. Examples of the scene descriptions are shown in **Fig. 3**.

3.2 Structure Input Text

In our proposed model, a question and a scene description are the input of the Transformer. The input string of the Transformer s is as below:

$$s = \text{[CLS]} + \text{description} + \text{[SEP]} + \text{question} + \text{[SEP]}, \quad (1)$$

where the description is a scene description generated from an image, and + represents a concatenation of sentences. An input string is generated for each question.

The string s is tokenized to obtain a sequence of n tokens $\mathbf{x} = [x_1, ..., x_n]$, and fed into a Transformer network to obtain a representation V that contains the information of the entire sequence.

$$\mathbf{x} = \text{tokenize}(s) \quad (2)$$

$$V = \text{Transformer}(\mathbf{x}). \quad (3)$$

3.3 Prediction

When making a prediction, the representation corresponding to the [CLS] token V^0 is fed into a classifier for the final joint representation f. Finally, the answer with the maximum score is selected as the final prediction

$$f = \text{Classifier}(V^0). \quad (4)$$

Table 1: Accuracies (%) on VQA-CP v2 test set.

Model	Feature Type	OverAll	Answer Type		
			Yes/No	Number	Other
Question-Only	Text	20.95	41.06	13.26	11.77
5 Captions	Text	36.19	45.54	16.66	36.43
UpDn [3]	Vector	37.94	42.27	11.93	46.05
RUBi [26]	Vector	44.23	64.85	11.83	44.11
LMH+CSS [27]	Vector	**58.95**	**84.37**	**49.42**	**48.21**
Scene Descriptions	Text	27.56	41.32	12.41	24.23
w/o actions	Text	27.37	41.08	**12.75**	24.20
w/o spacial	Text	27.74	41.40	12.63	24.52
w/o prepositions	Text	27.96	41.37	12.35	**25.11**
w/o non-actions	Text	**28.08**	**41.45**	12.32	25.09

4. Evaluation

4.1 Experimental Settings

We present experimental results on the VQA-CP v2 dataset [10], containing 658,111 questions about 219,158 images, of which the training set contains 438,183 questions for 120,932 images and the test set has 219,928 questions for 98,226 images. VQA-CP v2 has a different answer distribution for each question type to evaluate the model's robustness to question biases. Answers in the dataset are divided into three types: Yes/No, Number, and Other. The answer vocabulary is determined based on the number of occurrences of each unique answer in the dataset. Specifically, we include a phrase into the answer vocabulary if the phrase occurred nine times or more. This resulted in the vocabulary with 3,129 phrases. The standard performance metric for VQA is accuracy [1]. We followed this standard.

We use the BERT$_{\text{base}}$ uncased model [5,31] as our Transformer language model, which has 12 layers, 768 hidden sizes, 12 self-attention sizes, and 110 million parameters. This model does not differentiate uppercase and lowercase tokens. The maximum number of tokens per sequence is set to 128. Our classifier is a multi-layer perceptron (MLP) with two fully-connected (FC) layers. The ReLU activation function is inserted between the FC layers. The number of outputs is set to the size of the answer vocabulary. We use softmax cross entropy over the answer vocabulary for the loss function. We train our models with Adam optimizer [32]. We use the learning rate of 2×10^{-5} and the batch size of 128.

4.2 Results

Comparison with state-of-the-art In Table 1, we compare our approach against the state-of-the-art on VQA-CP v2. The state-of-the-art models are Bottom-Up and Top-Down Attention (UpDn) [3], RUBi [26], and Learned-Mixin+H+CSS (LMH+CSS) [27]. These state-of-the-art models all use deep visual features of images (denoted as Vector in Table 1). As for comparison with state-of-the-art, which uses deep visual features, our approach that uses scene descriptions as input of the Transformer (Scene Descriptions) reaches an average overall accuracy of 27.56%. This accuracy corresponds to a loss of −31.39 percentage points over the current state-of-the-art LMH+CSS [27]. It also corresponds to a loss of −10.40 percentage points over UpDn [3].

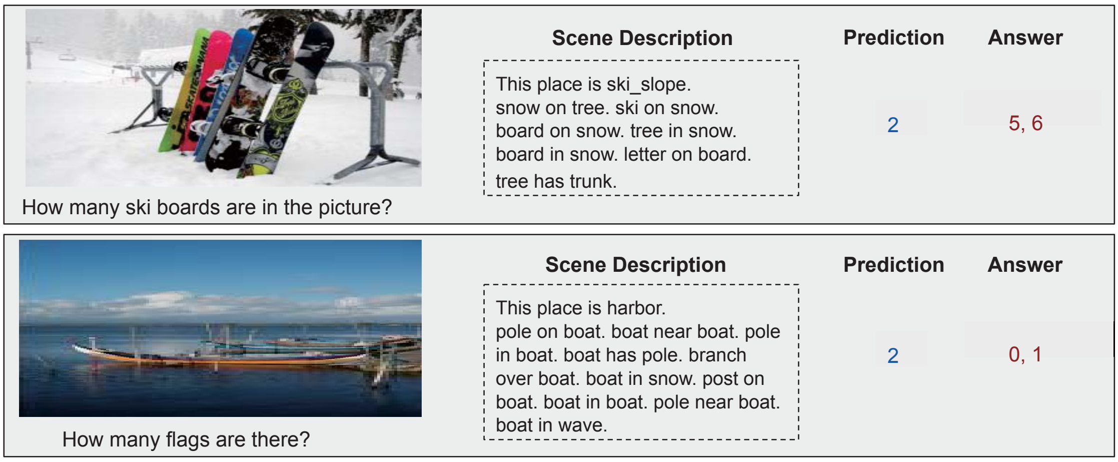
Comparison of Textual Representations For comparison, we also report the results of a model with the same architecture as the proposed model, but with only questions as input (Question-Only) and with five ground-truth captions from COCO dataset [33] attached to each image as input (5 Captions). In the same architecture, compared with Question-Only, Scene Descriptions reaches +6.61 percentage improvement, and also compared with 5 Captions, 5 Captions outperforms Scene Descriptions, which corresponds to a gain of +8.63 percentage. From these results, we can say that our scene descriptions are useful to answer the questions, but not so much than 5 Captions. Even so, the captions are using ground-truth data annotated by humans, while our scene descriptions are automatically generated. Hence, our scene descriptions have certain advantage over the captions.

Types of Predicate Comparison We manually divide predicate classes into four categories: actions (e.g. riding), non-actions (e.g. has), spatial (e.g. behind), and prepositions (e.g. with). To evaluate each category's impact, we remove one category out of four when generating scene descriptions and compare the results. The results show that we gain the best accuracy when removing predicates in the non-actions category. Conversely, the accuracy when removing predicates in the actions category is worst. From this observation, we found that predicates in the non-actions and prepositions have a negative effect. Hence, we may say that the input can be improved by cleaning the descriptions.

5. Discussion

Overall Comparison As the experimental results of the comparison with state-of-the-art show, state-of-the-art models outperform our proposed model that uses scene descriptions of images. This means that our textual representation, which is scene descriptions, is not competitive with CNN's deep visual features yet. Besides, comparing the result of Scene Descriptions with that of 5 Captions, it can be said that our scene descriptions are not enough to represent the images. Thus, it is needed to add or change the descriptions to describe the content of images more efficiently.

Evaluation with Answer Types The accuracy of our model for each answer type shows that the accuracy is the worst when the answer type is Number, which our model with scene descriptions suffer a performance drop from Question-Only. Some of other models also suffer from that kind of questions. We remove duplicates of triplets when we generate scene descriptions and don't



How many ski boards are in the picture?

Scene Description

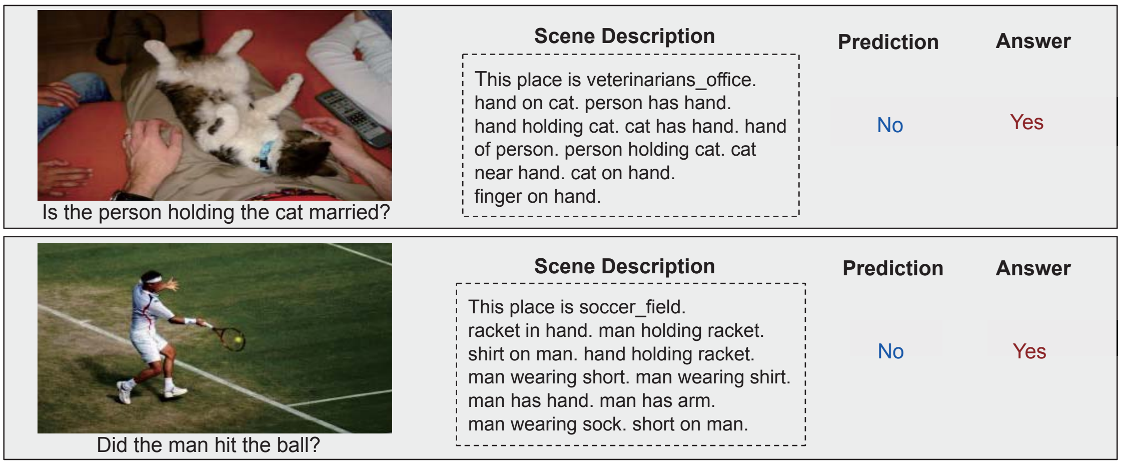
This place is ski_slope.
snow on tree. ski on snow.
board on snow. tree in snow.
board in snow. letter on board.
tree has trunk.

How many flags are there?

Scene Description

This place is harbor.
pole on boat. boat near boat. pole in boat. boat has pole. branch over boat. boat in snow. post on boat. boat in boat. pole near boat. boat in wave.

Fig. 4: Qualitative examples of questions which start with “How many ...”.



Is the person holding the cat married?

Scene Description

This place is veterinarians_office.
hand on cat. person has hand.
hand holding cat. cat has hand. hand of person. person holding cat. cat near hand. cat on hand.
finger on hand.

Did the man hit the ball?

Scene Description

This place is soccer_field.
racket in hand. man holding racket.
shirt on man. hand holding racket.
man wearing short. man wearing shirt.
man has hand. man has arm.
man wearing sock. short on man.

Fig. 5: Qualitative examples of questions which answer type is Yes/No.

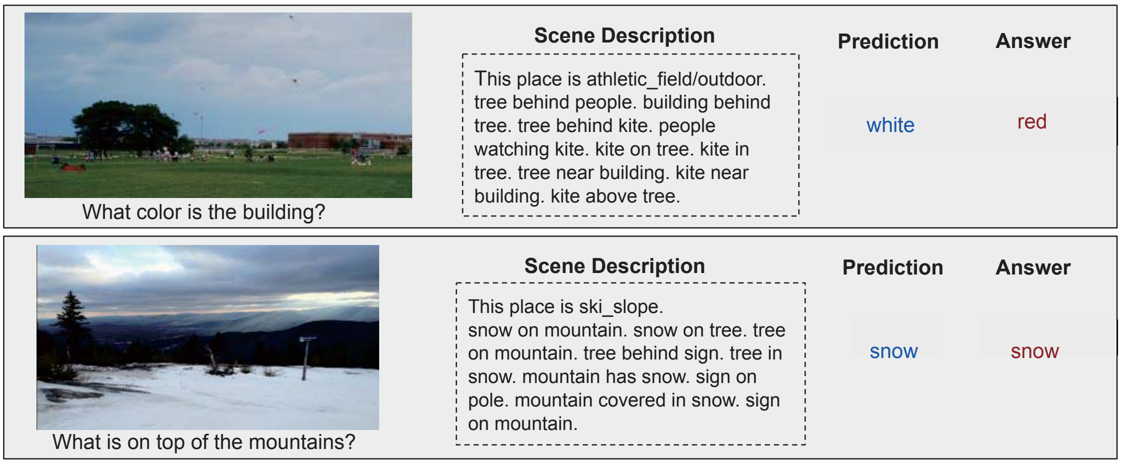
do anything to count objects in an image. Hence, it is reasonable that the result when the answer type is Number is bad. Note that number questions are still challenging problem, and even most state-of-the-art methods struggle to answer number questions.

The biggest difference between our model and LMH+CSS is the accuracy when the answer type is Yes/No. Our model’s accuracy is almost the same as when choosing an answer randomly, whereas LMH+CSS can correctly answer with over 80 % accuracy.

Qualitative Analysis To see how the generated scene descriptions work, we show some examples of successful and unsuccessful predictions of our model. In the VQA-CP v2 dataset, the correlations between question types and answers are very different in the train set than the test set. “2” is a common answer to “How many...” questions in the train set, but it is rare for such questions on the test set. However, as mentioned above, we do not do anything to count objects in an image, so our model learns to answer “2” when the question starts with “How many...”. In **Fig. 4**, we show examples of the failures when the questions begin with “How many...”. We find that our model’s prediction is “2”, regardless of the scene description or content of the image. In other words, our model is likely to have learned the bias of the dataset. We should think of the way to deal with this situation.

As we mentioned above, there is the biggest gap of accuracy between our model with scene descriptions and the state-of-the-art model when answering the questions whose answer type is Yes/No. In **Fig. 5**, we show some examples of unsuccessful predictions of our model when answer type is Yes/No. We find that generated scene descriptions miss some critical information to answer the questions correctly. For example, focusing on the example at the bottom, the question is, “Did the man hit the ball?” so we need to know the relationship between the man and the ball. However, there are not descriptions about the ball; thus, we do not have critical information to answer the question. From this observation, it can be said that the model tends to fail to answer correctly when a scene description is missing critical information, e.g., objects.

Additionally, in **Fig. 6**, we show some examples of success-



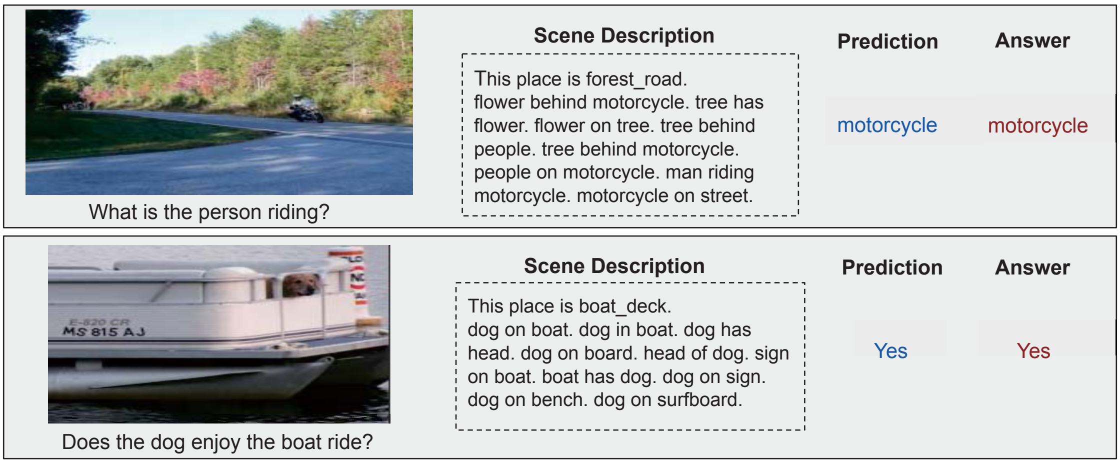
What color is the building?

This place is athletic_field/outdoor. tree behind people. building behind tree. tree behind kite. people watching kite. kite on tree. kite in tree. tree near building. kite near building. kite above tree.

What is on top of the mountains?

This place is ski_slope. snow on mountain. snow on tree. tree on mountain. tree behind sign. tree in snow. mountain has snow. sign on pole. mountain covered in snow. sign on mountain.

Fig. 6: Qualitative examples of questions which answer type is Other.



What is the person riding?

This place is forest_road. flower behind motorcycle. tree has flower. flower on tree. tree behind people. tree behind motorcycle. people on motorcycle. man riding motorcycle. motorcycle on street.

Does the dog enjoy the boat ride?

This place is boat_deck. dog on boat. dog in boat. dog has head. dog on board. head of dog. sign on boat. boat has dog. dog on sign. dog on bench. dog on surfboard.

Fig. 7: Qualitative examples of questions asking about actions in the images.

ful and unsuccessful predictions of our model when the answer type is Other. The example at the top is asking about the building's color, but we cannot generate descriptions of objects with attributes in our method. Hence, the model cannot answer the question correctly. On the other hand, in the example at the bottom, our model can accurately predict because there are some descriptions about the relationship between snow and the mountains. Hence, we can say that the model can answer the questions well when the scene descriptions contain the necessary information to answer the questions.

We also show the successful examples of our model in **Fig. 7**. In both examples, the questions ask about actions in the images, and our model's predictions are correct. This is because the scene descriptions can describe the actions in the images. For instance, looking at the example at the top, there is the description "man riding motorcycle" which contains the objects and their relation in the question. According to this fact, there is a possibility that textual descriptions can represent the relations between objects well. In other words, textual descriptions may be able to express the relationships more directly than deep visual features because they are a natural language.

6. Conclusion

In this paper, we proposed the method to generate scene descriptions of images and the model to use the generated descriptions as the input of the Transformer. The experimental results show that our model has not been competitive with state-of-the-art models that leverage deep visual features. However, through the qualitative analysis, we found that there is a possibility that textual descriptions can render the contents of images. In future work, we would like to explore the potential benefits of textual descriptions.

Acknowledgments

This work is partly supported by JSPS KAKENHI Grant No.18H03264.

References

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *CVPR*, 2015.

[2] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NeurIPS*, page 1682–1690, 2014.

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

[4] Tan Wang, Jianqiang Huang, Hanwang Zhang, and Qianru Sun. Visual Commonsense R-CNN. In *CVPR*, 2020.

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

[6] Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Large memory layers with product keys. *NeurIPS*, 2019.

[7] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *EMNLP Workshop BlackboxNLP*, pages 353–35, 2018.

[8] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392, 2016.

[9] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, 2016.

[10] Aishwarya Agrawal, Dhruv Batra, Devi Parikh, and Aniruddha Kembhavi. Don't just assume; look and answer: Overcoming priors for visual question answering. In *CVPR*, 2018.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[12] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 123(1):32–73, 2017.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, "2015".

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015.

[16] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. VilBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, pages 13–23, 2019.

[17] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

[18] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. UNITER: Universal image-text representation learning. In *ECCV*, 2020.

[19] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: Pre-training of generic visual-linguistic representations. In *ICLR*, 2020.

[20] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. VisualBERT: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.

[21] Noa Garcia and Yuta Nakashima. Knowledge-based video question answering with unsupervised scene descriptions. In *ECCV*, 2020.

[22] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. In *ECCV*, 2020.

[23] Jialin Wu, Zeyuan Hu, and Raymond Mooney. Generating question relevant captions to aid visual question answering. In *ACL*, pages 3585–3594, 2019.

[24] N. Ramachandran, Emmie Kehoe, and V. Sriram. Visual question answering via dense captioning. In *NeurIPS*, 2018.

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

[26] Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. RUBi: Reducing unimodal biases for visual question answering. In *NeurIPS*, pages 841–852, 2019.

[27] Long Chen, Xin Yan, Jun Xiao, Hanwang Zhang, Shiliang Pu, and Yueting Zhuang. Counterfactual samples synthesizing for robust visual question answering. In *CVPR*, pages 10800–10809, 2020.

[28] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. PAMI*, 2017.

[29] Ji Zhang, Yannis Kalantidis, Marcus Rohrbach, Manohar Paluri, Ahmed Elgammal, and Mohamed Elhoseiny. Large-scale visual relationship understanding. In *AAAI*, pages 9185–9194, 2019.

[30] Danfei Xu, Yuke Zhu, Christopher B Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *CVPR*, pages 5410–5419, 2017.

[31] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's Transformers: State-of-the-art natural language processing, 2020.

[32] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[33] Tsung-Yi Lin, Michael Maire, Serge 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.