Towards Visual Question Answering on Pathology Images

Xuehai He[*1], **Zhuo Cai**[*2], **Wenlan Wei**[3], **Yichen Zhang**[1], **Luntian Mou**[4], **Eric Xing**[5] **and Pengtao Xie**[1]
[1] UC San Diego, [2] Tsinghua University, [3] Wuhan University,
[4] Beijing University of Technology, [5] MBZUAI and CMU
p1xie@eng.ucsd.edu

Abstract

Pathology imaging is broadly used for identifying the causes and effects of diseases or injuries. Given a pathology image, being able to answer questions about the clinical findings contained in the image is very important for medical decision making. In this paper, we aim to develop a pathological visual question answering framework to analyze pathology images and answer medical questions related to these images. To build such a framework, we create PathVQA, a pathology VQA dataset with 32,795 questions asked from 4,998 pathology images. We also propose a three-level optimization framework which performs self-supervised pretraining and VQA finetuning end-to-end to learn powerful visual and textual representations jointly and automatically identifies and excludes noisy self-supervised examples from pretraining. We perform experiments on our created PathVQA dataset and the results demonstrate the effectiveness of our proposed methods. The datasets and code are available at https://github.com/UCSD-AI4H/PathVQA

1 Introduction

Pathology (Levison et al., 2012) studies the causes and effects of diseases or injuries. It underpins every aspect of patient care, such as diagnostic testing, providing treatment advice, preventing diseases using cutting-edge genetic technologies, to name a few. Given a pathology image, being able to answer questions about the clinical findings contained in the image is very important for medical decision-makings.

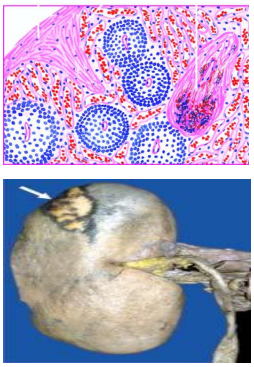
In this paper, we aim to develop a pathological visual question answering framework to analyze pathology images and answer medical questions related to these images. We first need to collect a dataset containing questions about pathology imaging. One possible way to create a pathology VQA dataset is crowdsourcing, which is used successfully for creating general domain VQA datasets (Malinowski and Fritz, 2014; Antol et al., 2015; Ren et al., 2015a; Johnson et al., 2017; Goyal et al., 2017). However, it is much more challenging to build medical VQA datasets than general domain VQA datasets via crowdsourcing. First, medical images such as pathology images are highly domain-specific, which can only be interpreted by well-educated medical professionals. It is rather difficult and expensive to hire medical professionals to help create medical VQA datasets. Second, to create a VQA dataset, one first needs to collect an image dataset. While images in the general domain are pervasive, medical images are very difficult to obtain due to privacy concerns.

To address these challenges, we resort to pathology textbooks, especially those that are freely accessible online, as well as online digital libraries. We extract images and captions from the textbooks and online digital libraries. Given these images, question-answer pairs are created based on image captions. These QA pairs are verified by medical professionals to ensure clinical meaningfulness and correctness. In the end, we created a pathology VQA dataset called PathVQA, which contains 32,795 questions asked from 4,998 pathology images. To our best knowledge, this is the first dataset for pathology VQA.

Given the pathology VQA dataset, the next step is to develop a pathology VQA system, which is also very challenging, due to the following reason. The medical concepts involved in PathVQA are very diverse while the number of question-answer pairs available for training is limited. Learning effective representations of these diverse medical concepts using limited data is technically difficult. Poorly learned representations lead to infe-

[*]Equal Contribution

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers), pages 708–718
August 1–6, 2021.



Q1: What are dilated and congested?
Q2: Are the sinuses dilated and congested?
Q3: Is there increased fibrosis in the red pulp, capsule and the trabeculae?
Q4: Where is increased fibrosis?
Q5: Is gamna-gandy body also seen?

Q1: What is slightly depressed on the surface?
Q2: Where is the wedge-shaped infarct slightly depressed?
Q3: Is the wedge-shaped infarct slightly depressed on the surface?
Q4: What is on the surface?
Q5: What is pale while the margin is haemorrhagic?

Figure 1: Two exemplar images with generated questions. Both images have three types of questions: "what", "where", and "yes/no".

rior VQA performance. To address this challenge, we propose a three-level optimization framework which performs cross-modal self-supervised pretraining (Tan and Bansal, 2019) and VQA finetuning of a pathology image encoder and a question encoder end-to-end to learn powerful visual and textual representations jointly and automatically identifies and excludes noisy self-supervised examples from pretraining. Experiments on our developed PathVQA dataset demonstrates the effectiveness of our proposed methods.

The major contributions of this paper are as follows:

- We create a pathology visual question answering dataset – PathVQA, to foster the research of medical VQA. To our best knowledge, this is the first dataset for pathology VQA.
- We propose a three-level optimization framework which performs cross-modal self-supervised pretraining and VQA finetuning of a pathology image encoder and a question encoder end-to-end to learn powerful visual and textual representations jointly and automatically identifies and excludes noisy self-supervised examples from pretraining.
- On our PathVQA dataset, we demonstrate the effectiveness of our proposed method.

2 Related Work

2.1 Medical VQA Datasets

To our best knowledge, there are two existing datasets for medical visual question answering. The VQA-Med (Abacha et al., 2019) dataset is created on 4,200 radiology images and has 15,292 question-answer pairs. Most of the questions are in multiple-choice (MC) style and can be answered by multi-way classifiers. This makes the difficulty of this dataset significantly lower. VQA-RAD (Lau et al., 2018) is a manually-crafted dataset where questions and answers are given by clinicians on radiology images. It has 3515 questions of 11 types. Our dataset differs from VQA-Med and VQA-RAD in two-fold. First, ours is about pathology while VQA-Med and VQA-RAD (Lau et al., 2018) are both about radiology. Second, our dataset is a truly challenging QA dataset where most of the questions are open-ended while in VQA-Med and VQA-RAD the majority of questions have a fixed number of candidate answers and can be answered by multi-way classification. Besides, the number of questions in our dataset is much larger than that in VQA-Med and VQA-RAD.

2.2 Cross-modal Self-supervised Learning

Cross-modal self-supervised learning learns representations for data with multiple modalities by solving cross-modal auxiliary tasks. VisualBERT (Li et al., 2019) learns representations for images and texts by implicitly aligning elements of a text and regions in an associated image with self-attention. CVLP (Shi et al., 2020) proposes an unbiased contrastive visual-linguistic pretraining approach, which constructs a self-supervised loss based on contrastive learning. ViLBERT (Lu et al., 2019) proposes to pretrain a vision-and-language BERT model through masked multi-modal modeling and alignment tasks, and then transfer the model to visual question answering tasks.

2.3 Data Selection and Data Reweighting

A number of approaches have been proposed for data selection. Ren et al. (2018) proposes a meta learning method to learn the weights of training examples by performing a meta gradient descent step on the weights of the current mini-batch of examples. Shu et al. (2019) propose a method which can adaptively learn an explicit weighting function directly from data.

3 The PathVQA Dataset

The PathVQA dataset consists of 32,795 question-answer pairs generated from 1,670 pathology images collected from two pathology textbooks: "Textbook of Pathology" (Muir et al., 1941) and

Table 1: Frequency of questions in different categories

Question type	Total number and percentage
Yes/No	16,329 (49.8%)
What	13,401 (40.9%)
Where	2,157 (6.6%)
How	595 (1.8%)
How much/many	139 (0.4%)
Why	114 (0.3%)
When	51 (0.2%)
Whose	9 (0.1%)

"Basic Pathology" (Robbins et al., 1981), and 3,328 pathology images collected from the PEIR[1] digital library. The question-answer pairs are generated using a semi-automated pipeline with linguistic rules. Figure 1 shows some examples.

On average, each image has 6.6 questions. The maximum and minimum number of questions for a single image is 14 and 1 respectively. The average number of words per question and per answer is 9.5 and 2.5 respectively. There are eight different categories of questions: what, where, when, whose, how, why, how much/how many, and yes/no. Table 1 shows the number of questions and percentage in each category. The questions in the first 7 categories are open-ended: 16,466 in total and accounting for 50.2% of all questions. The rest are close-ended "yes/no" questions. The questions cover various aspects of visual contents, including color, location, appearance, shape, etc. Such clinical diversity poses great challenges for AI models to solve this pathology VQA problem.

4 Method

We propose a three-level optimization based framework to perform VQA on PathVQA. In our framework, there are three learning stages, which are performed end-to-end jointly. In the first stage, self-supervised learning (He et al., 2019; Tan and Bansal, 2019) is performed to pretrain the image encoder and text encoder. In the second stage, we finetune the image encoder and text encoder on the PathVQA dataset. In the third stage, the trained model is validated on the validation set. In the first stage, we perform cross-modal self-supervised learning (Tan and Bansal, 2019) of an image encoder W and a text encoder T. The image encoder is used to extract visual features of pathology images. The text encoder is used to extract semantic features of questions and answers. Self-supervised learning (He et al., 2019) is an unsupervised representation learning approach where pretext tasks are defined solely based on the input data, and representations are learned by solving these pretext tasks.

There are many ways to construct pretext tasks. In our work, following (Tan and Bansal, 2019), we define a simple yet effective pretext task: in the PathVQA dataset, given a pathology image and a question, judge whether this question is about this image. From the PathVAQ training set D, we create another dataset $D' = \{(x_i, y_i, t_i)\}_{i=1}^{M}$ to perform the SSL task. There are M tuples, each containing a pathology image x from D and a question y from D. t_i is a binary variable where $t_i = 1$ if x and y are from the same training example in D and $t_i = 0$ if otherwise. Given D', we develop a model to map (x_i, y_i) to t_i. In this model, an image encoder is used to encode x_i and a text encoder is used to encode y_i; the concatenation of these two encodings is fed into a linear layer to predict whether the image matches with the question.

In self-supervised learning (He et al., 2019), the labels are typically constructed automatically without human supervision. As a result, they contain a lot of noises. For example, in D', t is determined simply based on whether x and y are from the training example in D. It is totally possible that a question y asked about an image x' is appropriate to be a question for another image x as well if x and x' are pathologically similar. In this case, the correct label t for (x, y) should be 1. However, it is set to 0 in D'. Training the encoders using these noisy and incorrect labels may confuse the encoders and result in poor-quality representations.

To address this problem, we aim to develop a method to automatically identify incorrectly auto-labeled examples in the training data of the SSL task. For each example (x, y, t) in D', we associate a selection variable $a \in [0, 1]$ with it. If a is close to 1, it means this example is correctly labeled; if a is close to 0, it means this example is incorrectly labeled. Let $l(f(x, y; W, T), t)$ denote the SSL loss defined on (x, y, t), where $f(x, y; W, T)$ is the predicted probability that $t = 1$ and $l(\cdot)$ is the cross-entropy loss. We multiply a with $l(f(x, y; W, T), t)$ so that if (x, y, t) is incorrectly

[1] http://peir.path.uab.edu/library/index.php?/category/2

labeled, its loss will be down-weighted to 0 and effectively (x, y, t) is excluded from the SSL pretraining process. In the end, only correctly-labeled examples are used for pretraining the encoders. To this end, in the first stage, we solve the following optimization problem:

$$W^*(A), T^*(A) = \underset{W,T}{\text{argmin}} \sum_{i=1}^{M} a_i l(f(x_i, y_i; W, T), t_i).$$

In this problem, the selection variables $A = \{a_i\}_{i=1}^M$ are fixed (we will discuss how to learn A later on). $\{a_i\}_{i=1}^M$ are used to weigh the losses of individual examples in D. W and T are trained by minimizing the sum of weighted losses. Note that the optimal solutions $W^*(A)$ and $T^*(A)$ are functions of A since $W^*(A)$ and $T^*(A)$ are functions of the loss function, which is a function of A.

In the second stage, we finetune the image encoder and text encoder in the VQA task defined on the PathVQA dataset D. Let V, U, R denote the network weights of the image encoder, text encoder, and QA network respectively. We train V, U, R by minimizing the VQA loss: $\sum_{i=1}^{N^{(\text{tr})}} L(d_i^{(\text{tr})}, V, U, R)$ where $d_i^{(\text{tr})}$ is a training example in D, consisting of an input pathology image, an input question, and an output answer. When training V and U, we encourage them to be close to the optimally trained network weights $W^*(A)$ and $T^*(A)$ of the image and text encoder in the first stage, to transfer the representations learned in the SSL task to the VQA task. The second stage amounts to solving the following optimization problem:

$$\begin{aligned} &V^*(W^*(A)), U^*(T^*(A)), R^* = \\ &\underset{V,U,R}{\text{argmin}} \sum_{i=1}^{N^{(\text{tr})}} L(d_i^{(\text{tr})}, V, U, R) + \\ &\gamma_1 \left\|V - W^*(A)\right\|_2^2 + \gamma_2 \left\|U - T^*(A)\right\|_2^2. \end{aligned} \quad (1)$$

where the L2 losses encourage V and U to be close to $W^*(A)$ and $T^*(A)$. γ_1 and γ_2 are tradeoff parameters. Note that $V^*(W^*(A))$ is a function of $W^*(A)$ since $V^*(W^*(A))$ is a function of $\|V - W^*(A)\|_2^2$ which is a function of $W^*(A)$. Similarly, $U^*(T^*(A))$ is a function of $T^*(A)$.

In the third stage, we apply the optimally trained VQA model including $V^*(W^*(A))$, $U^*(T^*(A))$, and R^* to make predictions on the validation dataset. Then we learn the selection variables A by minimizing the validation loss $\sum_{i=1}^{N^{(\text{val})}} L(d_i^{(\text{val})}, V^*(W^*(A)), U^*(T^*(A)), R^*)$.

Putting all these pieces together, we have the following three-level optimization framework:

$$\begin{aligned} &\min_A \sum_{i=1}^{N^{(\text{val})}} L(d_i^{(\text{val})}, V^*(W^*(A)), U^*(T^*(A)), R^*) \\ &s.t. \quad V^*(W^*(A)), U^*(T^*(A)), R^* = \\ &\quad \text{argmin}_{V,U,R} \sum_{i=1}^{N^{(\text{tr})}} L(d_i^{(\text{tr})}, V, U, R) \\ &\quad\quad +\gamma_1 \left\|V - W^*(A)\right\|_2^2 + \gamma_2 \left\|U - T^*(A)\right\|_2^2 \\ &\quad W^*(A), T^*(A) = \text{argmin}_{W,T} \sum_{i=1}^{M} a_i l(f(x_i, y_i; W, T), t_i) \end{aligned}$$

4.1 VQA Models

Our proposed method can be applied to any VQA method. In this work, we choose two well-established and state-of-the-art VQA methods to perform the study while noting that other VQA methods are applicable as well.

- **Method 1**: In (Tan and Bansal, 2019), a large-scale Transformer (Vaswani et al., 2017) model is built that consists of three encoders: an object relationship encoder, a language encoder, and a cross-modal encoder. The three encoders are built mostly based on two kinds of attention layers — self-attention layers and cross-attention layers. The object relationship encoder and the language encoder are both single-modality encoders. A cross-modal encoder is proposed to learn the connections between vision and language.

- **Method 2**: The method proposed in (Kim et al., 2018) uses a Gated Recurrent Unit (GRU) (Cho et al., 2014) recurrent network and a Faster R-CNN (Ren et al., 2015b) network to embed the question and the image. It extends the idea of co-attention to bilinear attention which considers every pair of multimodal channels.

5 Experiment

5.1 Experimental Settings

Data split We partition the images in the PathVQA dataset along with the associated questions into a training set, validation set, and testing set with a ratio of about 3:1:1. In the PathVQA dataset, the frequencies of question categories are imbalanced. Because of this, during the partition process, we perform sampling to ensure the frequencies of these categories in each set to be consistent. In the end, there are 19,755 question-answer pairs in the training set, 6,279 in the validation set, and 6,761 in the testing set.

Table 2: Accuracy (%), BLEU-n (%), and F1 (%) achieved by different methods. We denote cross-modal SSL on image-question pairs and image-answer pairs as CMSSL-IQ and CMSSL-IA.

Method	Accuracy	BLEU-1	BLEU-2	BLEU-3	F1
Method 1 without image	49.2	50.2	2.8	1.2	9.5
Method 1	57.6	57.4	3.1	1.3	9.9
Method 1 with CMSSL-IQ	58.7	59.0	3.5	2.1	11.0
Method 1 with CMSSL-IQ + three-level optimization framework	**63.4**	**63.7**	**4.1**	**2.5**	**12.2**
Method 1 with CMSSL-IA	58.6	58.9	3.4	2.0	10.3
Method 1 with CMSSL-IA + three-level optimization framework	62.4	62.2	3.6	2.3	12.0
Method 2 without image	46.2	46.5	1.0	0.0	0.8
Method 2	55.1	56.2	3.2	1.2	8.4
Method 2 with CMSSL-IQ	55.9	57.1	3.4	1.4	9.2
Method 2 with CMSSL-IQ + three-level optimization framework	**58.9**	**59.1**	3.8	**1.6**	9.2
Method 2 with CMSSL-IA	55.9	57.1	3.5	1.5	9.2
Method 2 with CMSSL-IA + three-level optimization framework	58.8	**59.1**	**4.0**	**1.6**	**9.4**

Evaluation metrics We perform evaluation using three metrics: 1) accuracy (Malinowski and Fritz, 2014) which measures the percentage of inferred answers that match exactly with the ground-truth using string matching; only exact matches are considered as correct; 2) macro-averaged F1 (Goutte and Gaussier, 2005), which measures the average overlap between the predicted answers and ground-truth, where the answers are treated as bag of tokens; 3) BLEU (Papineni et al., 2002), which measures the similarity of predicted answers and ground-truth by matching n-grams.

5.2 Results

Table 2 shows the VQA performance achieved by different methods. From this table, we make the following observations. **First**, for both Method 1 and Method 2, applying our three-level optimization based framework improves the performance. Our framework learns to identify and remove noisy and erroneous SSL training examples, which can avoid the model to be distorted by such bad-quality examples. **Second**, for both Method 1 and 2, applying cross-modal SSL (CMSSL) methods including CMSSL-IQ and CMSSL-IA improves the performance, which demonstrates the effectiveness of CMSSL. CMSSL uses auxiliary tasks, including judging whether an image matches with a question and judging whether an image matches with an answer, to learn semantic correspondence between image regions and words in questions/answers, which can improve the effectiveness of visual and textual representations for accurate VQA. It also learns image and text encoders by encourages the image and text encoders to solve auxiliary tasks, which reduces the risk of overfitting to the data-deficient VQA task on the small-sized training data.

One may suspect how much information in images is used during the inference of the answers? Could it be possible that the models simply learn the correlations between questions and answers and ignore the images? In light of these concerns, we perform studies where the images are not fed into VQA models and only questions are used as inputs for inferring answers. Table 2 shows the results of not using images ("Method 1/2 without image"). As can be seen, for both Method 1 and 2, ignoring images leads to substantial degradation of performance. This shows that images in our dataset provide valuable information for VQA and PathVQA is a meaningful VQA dataset. The models trained on our datasets are not degenerated to simply capturing the correlation between questions and answers.

6 Conclusion

In this paper, we build a pathology VQA dataset – PathVQA – that contains 32,795 question-answer pairs of 8 categories, generated from 4,998 images. Majority of questions in our dataset are open-ended, posing great challenges for the medical VQA research. Our dataset is publicly available. To address the challenges that the self-supervised training data may contain errors and the effective representations of pathology images and questions are difficult to learn on limited data, we propose a three-level optimization framework to automatically identify and remove problematic SSL training examples and learn sample-efficient visual and textual representations. Experiments on the PathVQA dataset demonstrate the effectiveness of our method.

Acknowledgement

This work is supported by gift funds from Tencent AI Lab and Amazon AWS.

References

Asma Ben Abacha, Sadid A Hasan, Vivek V Datla, Joey Liu, Dina Demner-Fushman, and Henning Müller. 2019. Vqa-med: Overview of the medical visual question answering task at imageclef 2019. In *CLEF2019 Working Notes.*

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *ICCV*.

Stanislaw Antol, C Lawrence Zitnick, and Devi Parikh. 2014. Zero-shot learning via visual abstraction. In *ECCV*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *HLT-NAACL workshop*.

Zhihao Fan, Zhongyu Wei, Piji Li, Yanyan Lan, and Xuanjing Huang. 2018. A question type driven framework to diversify visual question generation.

Cyril Goutte and Eric Gaussier. 2005. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In *European Conference on Information Retrieval*.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2019. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*.

Michael Heilman and Noah A Smith. 2009. Question generation via overgenerating transformations and ranking. Technical report.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. 2017. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*.

Aniruddha Kembhavi, Minjoon Seo, Dustin Schwenk, Jonghyun Choi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Are you smarter than a sixth grader? textbook question answering for multimodal machine comprehension. In *CVPR*.

Jin-Hwa Kim, Jaehyun Jun, and Byoung-Tak Zhang. 2018. Bilinear attention networks. In *NIPS*.

Diederik Kingma and Jimmy Ba. 2014a. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Diederik P Kingma and Jimmy Ba. 2014b. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *ACL*.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*.

Jason J Lau, Soumya Gayen, Asma Ben Abacha, and Dina Demner-Fushman. 2018. A dataset of clinically generated visual questions and answers about radiology images. *Scientific data*.

David Levison, Robin Reid, Alistair D Burt, David J Harrison, and Stewart Fleming. 2012. *Muir's textbook of pathology*. CRC Press.

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

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems*, pages 13–23.

Mateusz Malinowski and Mario Fritz. 2014. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *ACL*.

Robert Muir et al. 1941. Text-book of pathology. *Text-Book of Pathology.*, (Fifth Edition).

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Mengye Ren, Ryan Kiros, and Richard Zemel. 2015a. Image question answering: A visual semantic embedding model and a new dataset. *NIPS*.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015b. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Stanley L Robbins, Marcia Angell, and Vinay Kumar. 1981. *Basic pathology*. WB Saunders.

Lei Shi, Kai Shuang, Shijie Geng, Peng Su, Zhengkai Jiang, Peng Gao, Zuohui Fu, Gerard de Melo, and Sen Su. 2020. Contrastive visual-linguistic pretraining. *arXiv preprint arXiv:2007.13135*.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, pages 1919–1930.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *ECCV*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*.

Kristina Toutanova, Chris Brockett, Michael Gamon, Jagadeesh Jagarlamudi, Hisami Suzuki, and Lucy Vanderwende. 2007. The pythy summarization system: Microsoft research at duc 2007. In *Proc. of DUC*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2016. Stacked attention networks for image question answering. In *CVPR*.

C Lawrence Zitnick and Devi Parikh. 2013. Bringing semantics into focus using visual abstraction. In *CVPR*.

Table 3: Statistics of the data split.

	Training set	Validation set	Test set
# images	3,021	987	990
# QA pairs	19,755	6,279	6,761

Appendix

A Experimental setup

Table 3 shows dataset split statistics. We implement the methods using PyTorch and perform training on four GTX 1080Ti GPUs.

We basically follow the original model configurations used in (Tan and Bansal, 2019), (Kim et al., 2018), and (Yang et al., 2016). Data augmentation is applied to images, including shifting, scaling, and shearing. From questions and answers in the PathVQA dataset, we create a vocabulary of 4,631 words that have the highest frequencies.

In Method 1, we use the default hyperparameter settings in (Tan and Bansal, 2019). For the text encoder, the hidden size was set to 768. The image features were extracted from the outputs of the Faster-RCNN network, which is pretrained on BCCD[2] – a medical dataset containing blood cells photos, as well as on Visual Genome (Krishna et al., 2017). The initial learning rate was set to 5e-5 with the Adam (Kingma and Ba, 2014a) optimizer used. The batch size was set to 256. The model was trained for 200 epochs. In the SSL pretraining task on Method 1, we train a linear classifier with a dimension of 1,280 to judge whether an image matches with a question. In Method 2, words in questions and answers are represented using GloVe (Pennington et al., 2014) vectors pretrained on general-domain corpora such as Wikipedia, Twitter, etc. The image features are extracted from the outputs of the Faster-RCNN network pretrained on BCCD and Visual Genome. Given an image and a question, the model outputs an answer from a predefined set of answers. The dropout (Srivastava et al., 2014) rate for the linear mapping was set to 0.2 while for the classifier it was set to 0.5. The initial learning rate was set to 0.005 with the Adamax optimizer (Kingma and Ba, 2014b) used. The batch size was set to 512. The model was trained for 200 epochs. In the SSL pretraining task on Method 2, similar to that on Method 1, we train a linear classifier with a dimension of 1,280 to predict whether an image matches with a question. We optimize the selection variables using the Adam optimizer, with an initial learning rate of 0.01. We set γ_1 and γ_2 to 0.3 and 0.7 respectively.

[2]https://public.roboflow.ai/object-detection/bccd

B Dataset Creation

We develop a semi-automated pipeline to generate a pathology VQA dataset from pathology textbooks and online digital libraries. We manually check the automatically-generated question-answer pairs to fix grammar errors. The automated pipeline consists of two steps: (1) extracting pathology images and their captions from electronic pathology textbooks and the Pathology Education Informational Resource (PEIR) Digital Library[3] website; (2) generating questions-answer pairs from captions.

B.1 Extracting Pathology Images and Captions

Given a pathology textbook that is in the PDF format and available online publicly, we use two third-party tools PyPDF2[4] and PDFMiner[5] to extract images and the associated captions therefrom. PyPDF2 provides APIs to access the "Resources" object in each PDF page where the "XObject" gives information about images. PDFMiner allows one to obtain text along with its exact location in a page. To extract image captions from text in each page, we use regular expressions to search for snippets with prefixes of "Fig." or "Figure" followed by figure numbers and caption texts. For a page containing multiple images, we order them based on their locations; the same for the captions. Images and locations are matched based on their order. Given an online pathology digital library such as PEIR, we use two third-party tools Requests[6] and Beautiful Soup[7] to crawl images and the associated captions. Requests is an HTTP library built using Python and provides APIs to send HTTP/1.1 requests. Beautiful Soup generates the 'http.parser' and can access the urls and tags of the images on the website pages. Given a set of urls, we use Requests to read website pages and use Beautiful Soup to find images under the targeted HTML tags including the Content Division element $\langle div \rangle$, the unordered list element $\langle ul \rangle$, and the $\langle li \rangle$ element.

[3]http://peir.path.uab.edu/library/index.php?/category/2
[4]https://github.com/mstamy2/PyPDF2
[5]https://github.com/pdfminer/pdfminer.six
[6]https://requests.readthedocs.io/en/master/
[7]https://www.crummy.com/software/BeautifulSoup/

Table 4: Number of questions in different categories in each set

Dataset	Question types					
	What	Where	How	How much/many	Why	Yes/No
Training set	8083	1316	366	62	71	9804
Validation set	2565	409	108	21	21	3135
Testing set	2753	432	121	18	22	3390

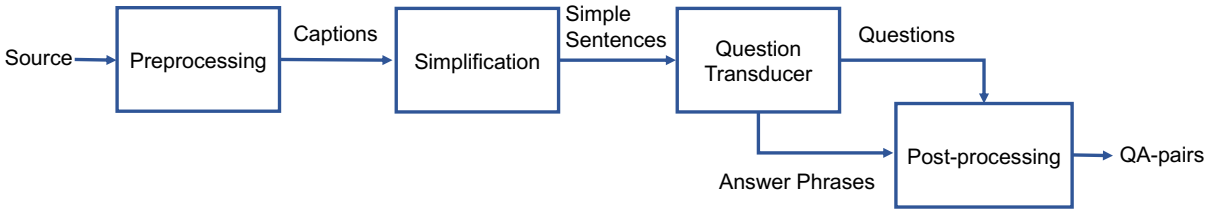


Figure 2: The framework of generating questions from captions

Then we can download images with Requests and write their captions directly to local files. Given the extracted image-caption pairs, we perform post-processing including (1) removing images that are not pathology images, such as flow charts and portraits; (2) correcting erroneous matching between images and captions.

B.2 Question Generation

In this section, we discuss how to semi-automatically generate questions from captions. Figure 2 shows the overall framework. We perform natural language processing of the captions using the Stanford CoreNLP (Klein and Manning, 2003) toolkit, including sentence split, tokenization, part-of-speech (POS) tagging, named entity recognition (NER), constituent parsing, and dependency parsing. Many sentences are long, with complicated syntactic structures. We perform sentence simplification to break a long sentence into several short ones. Given the subjects, verbs, clauses, etc. labeled by POS tagging and syntactic parsing, we rearrange them using the rules proposed in (Toutanova et al., 2007; Dorr et al., 2003) to achieve simplification. Figure 3 shows an example.

Given the POS tags and named entities of the simplified sentences, we generate questions for them: including "when"-type of questions for date and time entities and phrases such as "in/during ... stage/period", "before ...", and "after ..."; "how much/how many"-type of questions for words tagged as numbers; "whose" questions for possessive pronouns (e.g., "its", "their"); "where" questions for location entities and prepositional

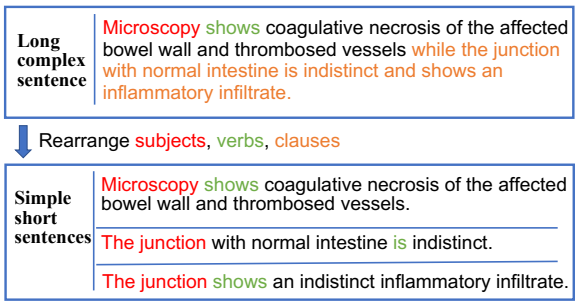


Figure 3: Sentence simplification

phrases starting with "inner", "within", "on the right/left of"; "how" questions for adjective words and phrases starting with "using", "via", "with", and "through", and "what" questions for the remaining noun phrases. Table 5 shows an example for each type of questions.

We use Tregex from Stanford CoreNLP tools (Manning et al., 2014), a tree query language including various relational operators based on the primitive relations of immediate dominance and immediate precedence, to implement the rules (Heilman and Smith, 2009) for transforming declarative sentences (captions) into questions.

To reduce grammatical errors, we avoid generating questions on sentences with adverbial clauses such as "chronic inflammation in the lung, showing all three characteristic histologic features". The question transducer mainly contains three steps. First, we perform the main verb decomposition based on the tense of the verb. For instance, we decompose "shows" to "does show". It is worth noting that for passive sentences with a structure of

Type	Original sentence	Question
What	**The end of the long bone** is expanded in the region of epiphysis.	What is expanded in the region of epiphysis?
Where	The left ventricle is **on the lower right** in this apical four-chamber view of the heart.	Where is the left ventricle in this apical four-chamber view of the heart?
When	**After 1 year of abstinence**, most scars are gone.	When are most scars gone?
How much/How many	**Two** multi-faceted gallstones are present in the lumen.	How many multi-faceted gallstones are present in the lumen?
Whose	The tumor cells and **their** nuclei are fairly uniform, giving a monotonous appearance.	The tumor cells and whose nuclei are fairly uniform, giving a monotonous appearance?
How	The trabecular bone forming the marrow space shows trabeculae **with osteoclastic activity at the margins**.	How does the trabecular bone forming the marrow space show trabeculae?

Table 5: Examples of generated questions for different types

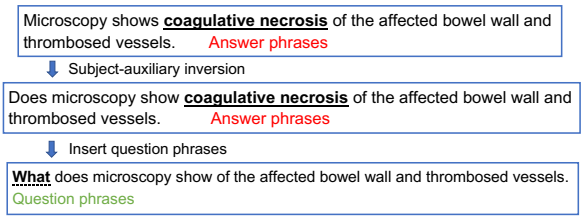


Figure 4: Syntactic transformation

"be+shown/presented/demonstrated", we keep their original forms rather than performing the verb decomposition. Second, we perform subject-auxiliary inversion. We invert the subject and the auxiliary verb in the declarative sentences to form the interrogative sentence. After the inversion, the binary "yes/no" questions are generated. For instance, as shown in Figure 4, the sentence "microscopy shows coagulative necrosis of the affected bowel wall and thrombosed vessels" is inverted to "does microscopy show coagulative necrosis of the affected bowel wall and thrombosed vessels?". To generate questions whose answers are "no", we randomly select a phrase with the same POS tagging from other captions to replace the head words in the original question. For example, we replace "coagulative necrosis" in the sentence "does microscopy show coagulative necrosis of the affected bowel wall and thrombosed vessels" with other noun phrases. Third, we remove the target answer phrases and insert the question phrase obtained previously to generate open-ended questions belonging to types of "what", "where", "when", "whose", "how", and "how much/how many" as shown in Table 5. For instance, we transduce "microscopy shows coagulative necrosis of the affected bowel wall and thrombosed vessels" to "what of the affected bowel wall and thrombosed vessels does microscopy show?" as shown in Figure 4. Given the automatically generated questions which may contain syntactic and semantic errors, we perform post-processing to fix those issues. We manually proofread all questions to correct misspellings, syntactic errors, and semantic inconsistencies. The questions and answers are further cleaned by removing extra spaces and irrelevant symbols. Questions that are too short or vague are removed. Articles appearing at the beginning of answers are stripped.

C Additional Related Works

C.1 VQA datasets

A number of visual question answering datasets have been developed in the general domain. DAQUAR (Malinowski and Fritz, 2014) is built on top of the NYU-Depth V2 dataset (Silberman et al., 2012) which contains RGBD images of indoor scenes. DAQUAR consists of (1) synthetic question-answer pairs that are automatically generated based on textual templates and (2) human-created question-answer pairs produced by five annotators. The VQA dataset (Antol et al., 2015) is developed on real images in MS COCO (Lin et al., 2014) and abstract scene images in (Antol et al., 2014; Zitnick and Parikh, 2013). The question-answer pairs are created by human annotators who are encouraged to ask "interesting" and "diverse" questions. VQA v2 (Goyal et al., 2017) is extended from the VQA (Antol et al., 2015) dataset to achieve more balance between visual and textual information, by collecting complementary images in a way that each question is associated with a pair of similar images with different answers. In the COCO-QA (Ren et al., 2015a) dataset, the question-answer pairs are automatically generated from image captions based on syntactic parsing and linguistic rules. CLEVR (Johnson et al., 2017; Kembhavi et al., 2017) is a dataset developed on rendered images of spatially related objects (including cube, sphere, and cylinder) with different sizes, materials, and colors. The locations and attributes of objects are annotated for each image. The questions are automatically generated from the annotations.

Table 6: Comparison of VQA datasets

	Domain	# images	# QA pairs	Answer type
DAQUAR	General	1,449	12,468	Open
VQA	General	204K	614K	Open/MC
VQA v2	General	204K	1.1M	Open/MC
COCO-QA	General	123K	118K	Open/MC
CLEVR	General	100K	999K	Open
VQA-Med	Medical	4,200	15,292	Open/MC
VQA-RAD	Medical	315	3,515	Open/MC
Ours	Medical	4,998	32,795	Open

The comparison of existing VQA datasets is shown in Table 6. The first five datasets are in the general domain while the last three are in the medical domain. Not surprisingly, the size of general-domain datasets (including the number of images and question-answer pairs) is much larger than that of medical datasets since general-domain images are much more available publicly and there are many qualified human annotators to generate QA pairs on general images. Our dataset is larger than the two medical datasets: VQA-Med and VQA-RAD, and majority of questions in our dataset are open-ended while majority of questions in VQA-Med and VQA-RAD are in multiple-choices style.

C.2 Automatic Construction of Question-Answer Pairs

Existing datasets have used automated methods for constructing question-answer pairs. In DAQUAR, questions are generated with templates, such as "How many {object} are in {image_id}?". These templates are instantiated with ground-truth facts from the database. In COCO-QA, the authors develop a question generation algorithm based on the Stanford syntactic parser (Klein and Manning, 2003), and they form four types of questions—"object", "number", "color", and "location" using hand-crafted rules. In CLEVR, the locations and attributes of objects in each image are fully annotated, based on which the questions are generated by an automated algorithm. Their algorithm cannot be applied to natural images where detailed annotation of objects and scenes are very difficult to obtain. In (Fan et al., 2018), the authors develop a conditional auto-encoder (Kingma and Welling, 2013) model to automatically generate questions from images. To train such a model, image-question pairs are needed, which incurs a chicken-and-egg problem: the goal is to generate questions, but realizing this goal needs generated questions. In VQA-Med, the authors collect medical images along with associated side information (e.g., captions, modalities) from the MedPix[8] database and generate question-answer pairs based on manually-defined patterns in (Lau et al., 2018).

D Number of questions in different categories for training, validation, and test set

For our data split, the number of questions in different categories in each set is shown in Table 4.

[8] https://medpix.nlm.nih.gov