

Learning to Read Chest X-Rays: Recurrent Neural Cascade Model for Automated Image Annotation

Hoo-Chang Shin¹, Kirk Roberts², Le Lu¹, Dina Demner-Fushman², Jianhua Yao¹, Ronald M Summers¹

¹Imaging Biomarkers and Computer-Aided Diagnosis Laboratory, Radiology and Imaging Sciences, Clinical Center

²Lister Hill National Center for Biomedical Communications, National Library of Medicine

National Institutes of Health, Bethesda, 20892-1182, USA

{hoochang.shin; le.lu; kirk.roberts; rms}@nih.gov; ddemner@mail.nih.gov; JYao@cc.nih.gov

Abstract

Despite the recent advances in automatically describing image contents, their applications have been mostly limited to image caption datasets containing natural images (e.g., Flickr 30k, MSCOCO). In this paper, we present a deep learning model to efficiently detect a disease from an image and annotate its contexts (e.g., location, severity and the affected organs). We employ a publicly available radiology dataset of chest x-rays and their reports, and use its image annotations to mine disease names to train convolutional neural networks (CNNs). In doing so, we adopt various regularization techniques to circumvent the large normal-vs-diseased cases bias. Recurrent neural networks (RNNs) are then trained to describe the contexts of a detected disease, based on the deep CNN features. Moreover, we introduce a novel approach to use the weights of the already trained pair of CNN/RNN on the domain-specific image/text dataset, to infer the joint image/text contexts for composite image labeling. Significantly improved image annotation results are demonstrated using the recurrent neural cascade model by taking the joint image/text contexts into account.

1. Introduction

Comprehensive image understanding requires more than single object classification. There have been many advances in automatic generation of image captions to describe image contents, which is closer to a more complete image understanding than classifying an image to a single object class. Our work is inspired by many of the recent progresses in image caption generation [44, 54, 36, 14, 61, 15, 6, 62, 31], as well as some of the earlier pioneering work [39, 17, 16]. The former have substantially

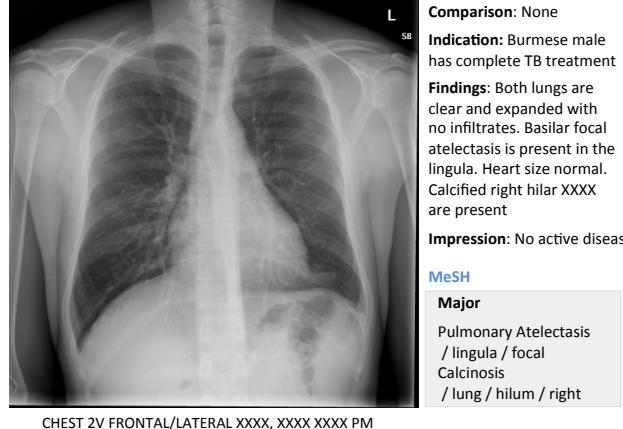


Figure 1. An example of OpenI [2] chest x-ray image, report, and annotations.

improved performance, largely due to the introduction of ImageNet database [13] and to advances in deep convolutional neural networks (CNNs), effectively learning to recognize the images with a large pool of hierarchical representations. Most recent work also adapt recurrent neural networks (RNNs), using the rich deep CNN features to generate image captions. However, the applications of the previous studies were limited to natural image caption datasets such as Flickr8k [25], Flickr30k [65], or MSCOCO [42] which can be generalized from ImageNet.

Likewise, there have been continuous efforts and progresses in the **automatic recognition and localization of specific diseases and organs**, mostly on datasets where target objects are explicitly annotated [50, 26, 55, 46, 40, 49]. Yet, learning from medical image text reports and generating annotations that describe diseases and their contexts

have been very limited. Nonetheless, providing a description of a medical image’s content similar to a radiologist would describe could have a great impact. A person can better understand a disease in an image if it is presented with its context, e.g., where the disease is, how severe it is, and which organ is affected. Furthermore, a large collection of medical images can be automatically annotated with the disease context and the images can be retrieved based on their context, with queries such as “*find me images with pulmonary disease in the upper right lobe*”.

In this work, we demonstrate how to automatically annotate chest x-rays with diseases along with describing the contexts of a disease, e.g., location, severity, and the affected organs. A publicly available radiology dataset is exploited which contains chest x-ray images and reports published on the Web as a part of the OpenI [2] open source literature and biomedical image collections. An example of a chest x-ray image, report, and annotations available on OpenI is shown in Figure 1.

A common challenge in medical image analysis is the data bias. When considering the whole population, diseased cases are much rarer than healthy cases, which is also the case in the chest x-ray dataset used. Normal cases account for 37% (2,696 images) of the entire dataset (7,284 images), compared to the most frequent disease¹ case “*opacity*” which accounts for 12% (840 images) and the next frequent “*cardiomegaly*” constituting for 9% (655 images). In order to circumvent the normal-vs-diseased cases bias, we adopt various regularization techniques in CNN training.

In analogy to the previous works using ImageNet-trained CNN features for image encoding and RNNs to generate image captions, we first train CNN models with one disease label per chest x-ray inferred from image annotations, e.g., “*calcified granuloma*”, or “*cardiomegaly*”. However, such single disease labels do not fully account for the context of a disease. For instance, “*calcified granuloma in right upper lobe*” would be labeled the same as the “*small calcified granuloma in left lung base*” or “*multiple calcified granuloma*”.

Inspired by the ideas introduced in [28, 64, 27, 62, 60], we employ the already trained RNNs to obtain the context of annotations, and recurrently use this to infer the image labels with contexts as attributes. Then we re-train CNNs with the obtained joint image/text contexts and generate annotations based on the new CNN features. With this recurrent cascade model, image/text contexts are taken into account for CNN training (images with “*calcified granuloma in right upper lobe*” and “*small calcified granuloma in left lung base*” will be assigned different labels), to ultimately generate better and more accurate image annotations.

¹Clinical findings, disorders, and other abnormal artifacts will be collectively referred to as *diseases* in this paper.

2. Related Work

This work was initially inspired by the early work in image caption generation [39, 17, 16], where we take more recently introduced ideas of using CNNs and RNNs [44, 54, 36, 14, 61, 15, 6, 62, 31] to combine recent advances in computer vision and machine translation. We also exploit the concepts of leveraging the mid-level RNN representations to infer image labels from the annotations [28, 64, 27, 62, 60].

Methods for mining and predicting labels from radiology images and reports were investigated in [51, 52, 57]. However, the image labels were mostly limited to disease names and did not contain much contextual information. Furthermore, the majority of cases in the datasets were diseased cases. In reality, most cases are normal, so that it is a challenge to detect relatively rarer diseased cases within such unbalanced data.

Mining images and image labels from a large collections of photo streams and blog posts on the Web were demonstrated in [34, 33, 35] where images could be searched with natural language queries. Associating neural word embeddings and deep image representations were explored in [37], but generating descriptions from such images/text pairs or image/word embeddings have not yet been demonstrated.

Detecting diseases from x-rays was demonstrated in [3, 45, 29], classifying chest x-ray image views in [63], and segmenting body parts in chest x-rays and computed tomography in [5, 21]. However, learning image contexts from text and re-generating image descriptions similar to what a human would describe have not yet been studied. To the best of our knowledge, this is the first study mining from a radiology image and report dataset, not only to classify and detect images but also to describe their context.

3. Dataset

We use a publicly available radiology dataset of chest x-rays and reports that is a subset of the OpenI [2] open source literature and biomedical image collections. It contains 3,955 radiology reports from the Indiana Network for Patient Care, and 7,470 associated chest x-rays from the hospitals’ picture archiving systems. The entire dataset has been fully anonymized via an aggressive anonymization scheme, which achieved 100% precision in de-identification. However, a few findings have been rendered uninterpretable. More details about the dataset and the anonymization procedure can be found in [11], and an example case of the dataset is shown in Figure 1.

Each report is structured as *comparison*, *indication*, *findings*, and *impression* sections, in line with a common radiology reporting format for diagnostic chest x-rays. In the example shown in Figure 1, we observe an error resulting from the aggressive automated de-identification scheme. A

MeSH Term	Total	Overlap	Overlap Percent
normal	2,696	0	0%
opacity	840	666	79%
cardiomegaly	655	492	75%
calcinosis	558	444	80%
lung/hypoventilation	539	361	67%
calcified granuloma	511	303	59%
thoracic vertebrae/degenerative	471	296	63%
lung/hyperdistension	400	260	65%
spine/degenerative	337	219	65%
catheters, indwelling	222	159	72%
granulomatous disease	213	165	78%
nodule	211	160	76%
surgical instruments	180	120	67%

Table 1. Thirteen most frequent MeSH terms appearing over 180 times, and the number of the terms mentioned with other terms (overlap) in an image and their percentages.

word possibly indicating a disease was falsely detected as a personal information, and was thereby “anonymized” as “XXXX”. While radiology reports contain comprehensive information about the image and the patient, they may also contain information that cannot be inferred from the image content. For instance, in the example shown in Figure 1, it is probably impossible to determine that the image is of a Burmese male.

On the other hand, a manual annotation of MEDLINE® citations with controlled vocabulary terms (Medical Subject Headings (MeSH®) [1]) is known to significantly improve the quality of the image retrieval results [20, 22, 10]. MeSH terms for each radiology report in OpenI (available for public use) are annotated according to the process described in [12]. We use these to train our model.

Nonetheless, it is impossible to assign a single image label based on MeSH and train a CNN to reproduce them, because MeSH terms seldom appear individually when describing an image. The twenty most frequent MeSH terms appear with other terms in more than 60% of the cases. Normal cases (term “*normal*”) on the contrary, do not have any overlap, and account for 37% of the entire dataset. The thirteen most frequent MeSH terms appearing more than 180 times are provided in Table 1, along with the total number of cases in which they appear, the number of cases they overlap with in an image and the overlap percentages. The x-ray images are provided in Portable Network Graphics (PNG) format, with sizes varying from 512×420 to 512×624 . We rescale all CNN input training and testing images to a size of 256×256 .

4. Disease Label Mining

The CNN-RNN based image caption generation approaches [44, 54, 36, 14, 61, 15, 6, 62, 31] require a well-trained CNN to encode input images effectively. Unlike natural images that can simply be encoded by ImageNet-

trained CNNs, chest x-rays differ significantly from the ImageNet images. In order to train CNNs with chest x-ray images, we sample some frequent annotation patterns **with less overlaps for each image**, in order to assign image labels to each chest x-ray image and train with cross-entropy criteria. This is similar to the previous works from [51, 52, 57], which mines disease labels of images from their annotation text (radiology reports).

We find 17 unique patterns of MeSH term combinations appearing in 30 or more cases. This allows us to split the dataset in training/validation/testing cases as 80%/10%/10% and place at least 10 cases each in the validation and testing sets. They include the terms shown in Table 1, as well as *scoliosis*, *osteophyte*, *spondylosis*, *fractures/bone*. MeSH terms appearing frequently but without unique appearance patterns include *pulmonary atelectasis*, *aorta/tortuous*, *pleural effusion*, *cicatrix*, etc. They often appear with other disease terms (e.g. *consolidation*, *airspace disease*, *atherosclerosis*). We retain about 40% of the full dataset with this disease image label mining, where the annotations for the remaining 60% of images are more complex (and it is therefore difficult to assign a single disease label).

5. Image Classification with CNN

We use the aforementioned 17 unique disease annotation patterns (in Table 1, and *scoliosis*, *osteophyte*, *spondylosis*, *fractures/bone*) to label the images and train CNNs. For this purpose, we adopt various regularization techniques to deal with the **normal-vs-diseased cases bias**. For our default CNN model we chose the simple yet effective Network-In-Network (NIN) [41] model because the model is small in size, fast to train, and achieves similar or better performance to the most commonly used AlexNet model [38]. We then test whether our data can benefit from a more complex state-of-the-art CNN model, i.e. GoogLeNet [58].

From the 17 chosen disease annotation patterns, **normal cases account for 71% of all images**, well above the numbers of cases for the remaining 16 disease annotation patterns. **We balance the number of samples for each case by augmenting the training images of the smaller cases where we randomly crop 224×224 size images from the original 256×256 size image.**

5.1. Regularization by Batch Normalization and Data Dropout

Even when we balance the dataset by augmenting many diseased samples, it is difficult for a CNN to learn a good model to distinguish many diseased cases from normal cases which have many variations on their original samples. It was shown in [27] that normalizing via **mini-batch statistics during training** can serve as an effective regularization technique to improve the performance of a CNN model. By

	training accuracy	validation accuracy
NIN with batch-normalization (BN)	94.06%	56.65%
NIN with data-dropout (DDropout)	98.78%	58.38%
NIN with BN and DDropout	100.0%	62.43%

Table 2. Training and validation accuracy of NIN model with batch-normalization, data-dropout, and both. Diseased cases are very limited compared to normal cases, leading to overfitting even with regularizations.

	training accuracy	validation accuracy
GoogLeNet with BN and DDropout	98.11%	66.40%
GoogLeNet with BN, DDropout, No-Crop	100.0%	69.84%

Table 3. Training and validation accuracy of GoogLeNet model with batch-normalization, data-dropout, and without cropping the images for data augmentation.

normalizing via mini-batch statistics, the training network was shown not to produce deterministic values for a given training example, thereby regularizing the model to generalize better.

Inspired by this and by the concept of Dropout [23], we regularize the normal-vs-diseased cases bias via randomly dropping out an excessive proportion of normal cases compared to the frequent diseased pattern when sampling mini-batches. We then normalize according to the mini-batch statistics where each mini-batch consists of a balanced number of samples per disease case and a random selection of normal case samples. The number of samples for disease cases is balanced by random cropping during training, where each image of a diseased case is augmented at least four times.

We test both regularization techniques to assess their effectiveness on our dataset. The training and validation accuracies of the NIN model with batch-normalization, data-dropout, and both are provided in Table 2. While batch-normalization and data-dropout alone do not significantly improve performance, combining both increases the validation accuracy by about 2%.

5.2. Effect of Model Complexity

We also validate whether the dataset can benefit from a more complex GoogLeNet [58], which is arguably the current state-of-the-art CNN architecture. We apply both batch-normalization and data-dropout, and follow recommendations suggested in [27] (where human accuracy on the ImageNet dataset is superseded): increase learning rate, remove dropout, remove local response normalization. The final training and validation accuracies using GoogLeNet model are provided in Table 3, where we achieve a higher ($\sim 4\%$) accuracy². We also observe a further $\sim 3\%$ in-

²The final models are trained with default learning rate of 1.0, with step down learning rate scheduling decreasing the learning rate by 50% and 33% each for NIN and GoogLeNet model in every 1/3th of the total 100

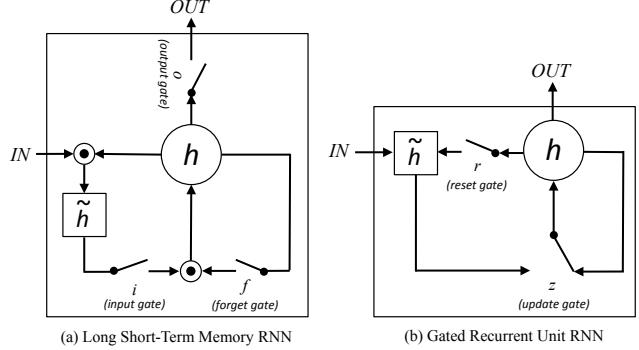


Figure 2. Simplified illustrations of (a) Long Short-Term Memory (LSTM) and (b) Gated Recurrent Unit (GRU) RNNs. The illustrations are adapted and modified from Figure 2 in [8].

crease in accuracy when the images are no longer cropped, but merely duplicated to balance the dataset.

6. Annotation Generation with RNN

We use recurrent neural networks (RNNs) to learn the annotation sequence given input image CNN embeddings. We test both Long Short-Term Memory (LSTM) [24] and Gated Recurrent Unit (GRU) [7] implementations of RNNs. Simplified illustrations of LSTM and GRU are shown in Figure 2, and the details of both RNN implementations are briefly introduced below.

6.1. Recurrent Neural Network Implementations

Long Short-Term Memory The LSTM implementation, originally proposed in [24], has been successfully applied to speech recognition [19], sequence generation [18], machine translation [7, 56, 43], and several image caption generation works mentioned in the main paper. LSTM is defined by the following equations:

$$i_t = \sigma(W^{(i)}x_t + U^{(i)}m_{t-1}) \quad (1)$$

$$f_t = \sigma(W^{(f)}x_t + U^{(f)}m_{t-1}) \quad (2)$$

$$o_t = \sigma(W^{(o)}x_t + U^{(o)}m_{t-1}) \quad (3)$$

$$\tilde{h}_t = \tanh(W^{(h)}x_t + U^{(h)}m_{t-1}) \quad (4)$$

$$h_t = f_t \odot h_{t-1} + i_t \odot \tilde{h}_t \quad (5)$$

$$m_t = o_t \odot \tanh(h_t) \quad (6)$$

where i_t is the input gate, f_t the forget gate, o_t the output gate, h_t the state vector (memory), \tilde{h}_t the new state vector (new memory), and m_t the output vector. W is a matrix of trained parameters (weights), and σ is the logistic sigmoid

training epochs. We could not achieve high enough validation accuracy using exponential learning rate decay as in [27].

function. \odot represents the product of a vector with a gate value.

Please note that the notation used for the memory (h_t , \tilde{h}_t) and output (m_t) vectors differs from that in [24] and the other previous work. Our notation is intended to simplify the annotations and figures comparing LSTM to GRU.

Gated Recurrent Unit The GRU implementation has been proposed most recently in [7], where it was successfully applied to machine translation. GRU is defined by the following equations:

$$z_t = \sigma(W^{(z)}x_t + U^{(z)}h_{t-1}) \quad (7)$$

$$r_t = \sigma(W^{(r)}x_t + U^{(r)}h_{t-1}) \quad (8)$$

$$\tilde{h}_t = \tanh(Wx_t + r_t \odot Uh_{t-1}) \quad (9)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \quad (10)$$

where z_t is the update gate, r_t the reset gate, \tilde{h}_t the new state vector, and h_t the final state vector.

6.2. Training

The number of MeSH terms describing diseases ranges from 1 to 8 (except *normal* which is one word), with a mean of 2.56 and standard deviation of 1.36. The majority of descriptions contain up to five words. Since only 9 cases have images with descriptions longer than 6 words, we ignore these by constraining the RNNs to unroll up to 5 time steps. We zero-pad annotations with less than five words with the end-of-sentence token to fill in the five word space.

The parameters of the gates in LSTM and GRU decide whether to update their current state \mathbf{h} to the new candidate state $\tilde{\mathbf{h}}$, where these are learned from the previous input sequences. Further details about LSTM can be found in [24, 15, 14, 61], and about GRU and its comparisons to LSTM in [7, 30, 9, 8, 32]. We set the initial state of RNNs as the CNN image embedding ($\text{CNN}(I)$), and the first annotation word as the initial input. The output of the RNNs are the following annotation word sequences, and we train RNNs by minimizing the negative log likelihood of output sequences and true sequences:

$$L(I, S) = - \sum_{t=1}^N \{P_{\text{RNN}}(y_t = s_t) | \text{CNN}(I)\}, \quad (11)$$

where y_t is the output word of RNN in time step t , s_t the correct word, $\text{CNN}(I)$ the CNN embedding of input image I , and N the number of words in the annotation ($N = 5$ with the end-of-sequence zero-padding). Equation 11 is not a true conditional probability (because we only initialize the

	train	validation	test
	BLEU -1/-2/-3/-4	BLEU -1/-2/-3/-4	BLEU -1/-2/-3/-4
LSTM	82.6 / 19.2 / 2.2 / 0.0	67.4 / 15.1 / 1.6 / 0.0	78.3 / 0.4 / 0.0 / 0.0
GRU	98.9 / 46.9 / 1.2 / 0.0	85.8 / 14.1 / 0.0 / 0.0	75.4 / 3.0 / 0.0 / 0.0

Table 4. BLEU scores validated on the training, validation, test set, using LSTM and GRU RNN models for the sequence generation.

RNNs' state vector to be $\text{CNN}(I)$ but a convenient way to describe the training procedure.

Unlike the previous work [31, 15, 14] where they use the last (FC-8) or second last (FC-7) fully-connected layers of AlexNet [38] or VGG-Net [53] model, the NIN or GoogLeNet models replace the fully-connected layers with the average-pooling layers [41, 58]. We therefore use the output of the last spatial average-pooling layer as the image embedding to initialize the RNN state vectors. The size of our RNNs' state vectors are $\mathbb{R}^{1 \times 1024}$, which is identical to the output size of the average-pooling layers from NIN and GoogLeNet.

6.3. Sampling

In sampling we again initialize the RNN state vectors with the CNN image embedding ($\mathbf{h}_{t=1} = \text{CNN}(I)$). We then use the CNN prediction of the input image as the first word as the input to the RNN, to sample following sequences up to five words. As previously, images are normalized by the batch statistics before being fed to the CNN. We use GoogLeNet as our default CNN model since it performs better than the NIN model in Sec. 5.2.

6.4. Evaluation

We evaluate the annotation generation on the BLEU [47] score averaged over all of the images and their annotations in the training, validation, and test set. BLEU scores is a metric measuring a modified form of precision to compare n-gram words of generated and reference sentences. The BLEU-N scores evaluated are provided in Table 4. The BLEU-N scores are evaluated for cases with $\geq N$ words in the annotations, using the implementation of [4].

We noticed that LSTM is easier to train, while GRU model yields better results with more carefully selected hyper-parameters³. While we find it difficult to conclude which model is better, the GRU model seems to achieve higher scores on average.

³The final LSTM models are obtained with – learning rate: 2×10^{-3} , learning rate decay: 0.97, decay rate: 0.95, without dropout; and the final GRU model is obtained with – learning rate: 1×10^{-4} , learning rate decay: 0.99, decay rate: 0.99, with dropout rate: 0.9. With the same setting, adding dropout to LSTM model has adverse effect on its validation loss, similarly when increasing the number of LSTM layers to 3. The number of layers are 2 for both RNN models, and they are both trained with the batch size of 50.

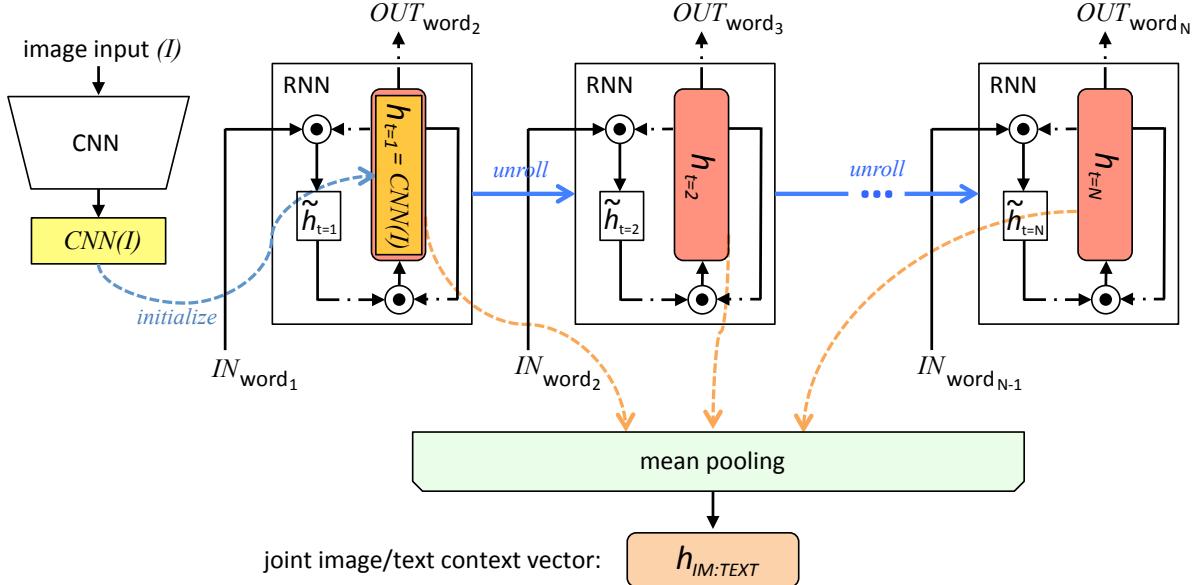


Figure 3. An illustration of how joint image/text context vector is obtained. RNN’s state vector (h) is initialized with the CNN image embedding ($\text{CNN}(I)$), and it’s unrolled over the annotation sequences with the words as input. Mean-pooling is applied over the state vectors in each word of the sequence, to obtain the joint image/text context vector. All RNNs share the same parameters, which are trained in the first round.

7. Recurrent Cascade Model for Image Labeling with Joint Image/Text Context

In Section 5, our CNN models are trained with disease labels only where the context of diseases are not considered. For instance, the same *calcified granuloma* label is assigned to all image cases that actually may describe the disease differently in a finer semantic level, such as “*calcified granuloma in right upper lobe*”, “*small calcified granuloma in left lung base*”, and “*multiple calcified granuloma*”.

Meanwhile, the RNNs trained in Section 6 encode the text annotation sequences given the CNN embedding of the image the annotation is describing. We therefore use the already trained CNN and RNN to infer better image labels, integrating the contexts of the image annotations beyond just the name of the disease. This is achieved by generating *joint image/text context vectors* that are computed by applying mean-pooling on the state vectors (h) of RNN at each step over the annotation sequence. Note, that the state vector of RNN is initialized with the CNN image embeddings ($\text{CNN}(I)$), and the RNN is unrolled over the annotation sequence, taking each word of the annotation as input. The procedure is illustrated in Figure 3, and the RNNs share the same parameters.

The obtained *joint image/text context vector* ($h_{\text{im:text}}$) encodes the image context as well as the text context describing the image. Using a notation similar to Equation 11, the *joint image/text context vector* can be written as:

$$h_{\text{im:text}} = \frac{\sum_{t=1}^N \{h_{\text{RNN}}(x_t) | \text{CNN}(I)\}}{N}, \quad (12)$$

where x_t is the input word in the annotation sequence with N words. Different annotations describing a disease are thereby separated into different categories by the $h_{\text{im:text}}$, as shown in Figure 4. In Figure 4, the $h_{\text{im:text}}$ vectors of about fifty annotations describing *calcified granuloma* are projected onto two-dimensional planes via dimensionality reduction ($\mathbb{R}^{1 \times 1024} \rightarrow \mathbb{R}^{1 \times 2}$), using the t-SNE [59] implementation of [48]. We use the GRU implementation of the RNN because it showed better overall BLEU scores in Table 4. A visualization example for the annotations describing *opacity* can be found in the supplementary material.

From the $h_{\text{im:text}}$ generated for each of the image/annotation pair in the training and validation sets, we obtain new image labels taking disease context into account. In addition, we are no longer limited to disease annotation mostly describing a single disease. The joint image/text context vector $h_{\text{im:text}}$ summarizes both the image’s context and word sequence, so that annotations such as “*calcified granuloma in right upper lobe*”, “*small calcified granuloma in left lung base*”, and “*multiple calcified granuloma*” have different vectors based on their contexts.

Additionally, disease labels used in Section 5 with unique annotation patterns now have more cases, as cases with a disease described by different annotation words are no longer filtered out. For example, *calcified granuloma* previously had only 139 cases because cases with multiple



Figure 4. Visualization of joint image/text context vectors of about 50 samples of the annotations describing disease *calcified granuloma* on 2D planes. Dimension reduction ($\mathbb{R}^{1 \times 1024} \rightarrow \mathbb{R}^{1 \times 2}$) is performed using t-SNE [59]. Annotations describing a same disease can be divided into different “labels” based on their joint image/text contexts.

diseases mentioned or with long description sequences were filtered out. At present, 414 cases are associated with *calcified granuloma*. Likewise, opacity now has 207 cases, as opposed to the previous 65. The average number of cases all first-mentioned disease labels has is 83.89, with a standard deviation of 86.07, a maximum of 414 (*calcified granuloma*) and a minimum of 18 (*emphysema*).

For a disease label having more than 170 cases ($n \geq 170 = (\text{average} + \text{standard deviation})$), we divide the cases into sub-groups of more than 50 cases by applying *k-means* clustering to the $h_{\text{im:text}}$ vector with $k = \text{Round}(n/50)$. We train the CNN once more with the additional labels (57, compared to 17 in Section 5), train the RNN with the new CNN image embedding, and finally generate image annotations. The new RNN training cost function (compared to Equation 11) can be expressed as:

$$L(I, S) = - \sum_{t=1}^N [P_{\text{RNN}_{\text{iter}=1}}(y_t = s_t) \mid \{\text{CNN}_{\text{iter}=1}(I) \mid h_{\text{im:text}_{\text{iter}=0}}\}], \quad (13)$$

where $h_{\text{im:text}_{\text{iter}=0}}$ denotes the joint image/text context vector obtained from the first round (with limited cases and im-

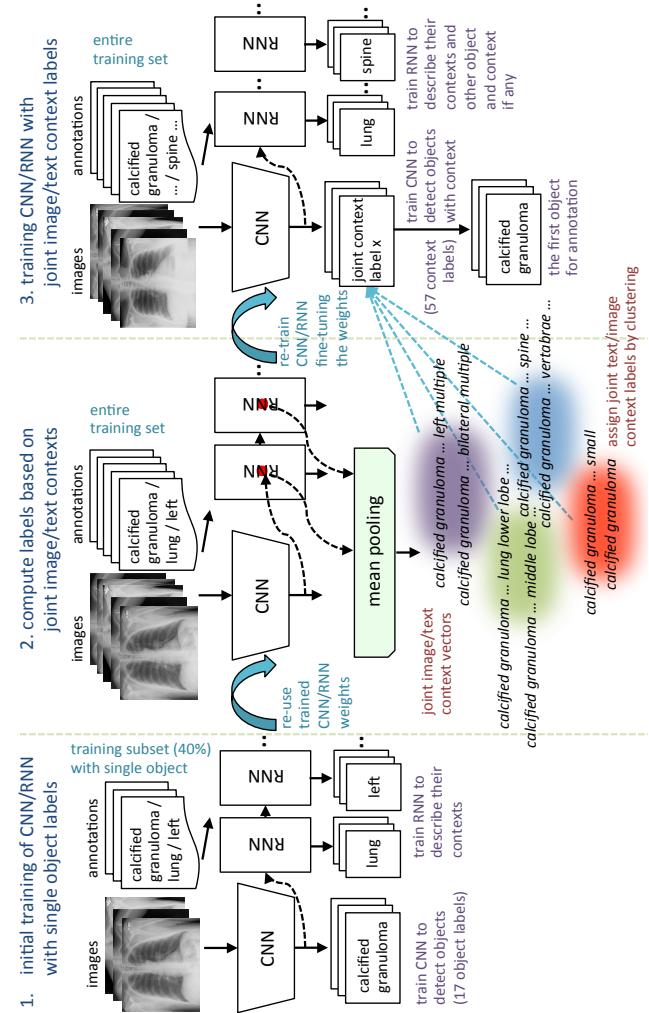


Figure 5. Overall workflow for the automated image annotation.

age labels at *0th iteration*) of CNN and RNN training. In the second CNN training round (*1st iteration*), we fine-tune from the previous $\text{CNN}_{\text{iter}=0}$, by replacing the last classification layer with the new set of labels ($17 \rightarrow 57$) and training it with a lower learning rate (0.1), except for the classification layer. The overall workflow is illustrated in Figure 5.

7.1. Evaluation

The final evaluated BLEU scores are provided in Table 5. We achieve better overall BLEU scores than those in Table 4 before using the joint image/text context. It is noticeable that higher BLEU-N ($N > 1$) scores are achieved compared to Table 4, indicating that more comprehensive image contexts are taken into account for the CNN/RNN training. Also, slightly better BLEU scores are obtained using GRU on average and higher BLEU-1 scores are acquired using LSTM, although the comparison is empirical. Examples of

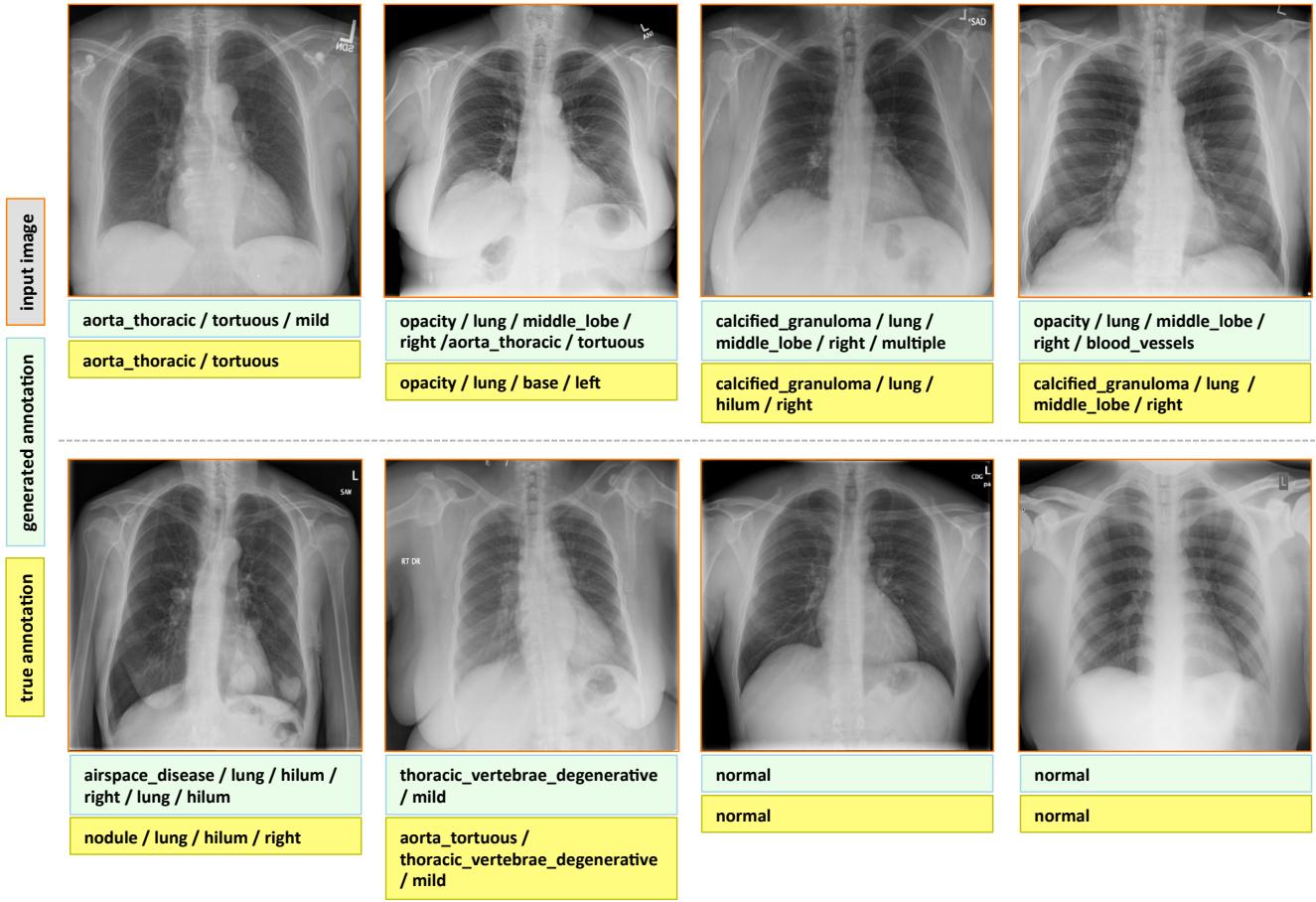


Figure 6. Examples of annotation generations (light green box) compared to true annotations (yellow box) for input images in the test set.

	train	validation	test
	BLEU -1/-2/-3/-4	BLEU -1/-2/-3/-4	BLEU -1/-2/-3/-4
LSTM	97.2 / 67.1 / 14.9 / 2.8	68.1 / 30.1 / 5.2 / 1.1	79.3 / 9.1 / 0.0 / 0.0
GRU	89.7 / 61.7 / 28.5 / 11.0	61.9 / 29.6 / 11.3 / 2.0	78.5 / 14.4 / 4.7 / 0.0

Table 5. BLEU scores validated on the training, validation, test set, using LSTM and GRU RNN models trained on the first iteration, for the sequence generation.

generated annotations on the chest x-ray images are shown in Figure 6. These are generated using the GRU model, and more examples can be found in the supplementary material.

8. Conclusion

We present an effective framework to learn, detect disease, and describe their contexts from the patient chest x-rays and their accompanying radiology reports with Medical Subject Headings (MeSH) annotations. Furthermore, we introduce an approach to mine joint contexts from a collection of images and their accompanying text, by summarizing the CNN/RNN outputs and their states on each of the image/text instances. Higher performance on text genera-

tion is achieved on the test set if the joint image/text contexts are exploited to re-label the images and to train the proposed CNN/RNN framework subsequently.

To the best of our knowledge, this is the first study that mines from a publicly available radiology image and report dataset, not only to classify and detect disease in images but also to describe their context similar to a human observer would read. While we only demonstrate on a medical dataset, the suggested approach could also be applied to other application scenario with datasets containing co-existing pairs of images and text annotations, where the domain-specific images differ from those of the ImageNet.

Acknowledgement

This work was supported in part by the Intramural Research Program of the National Institutes of Health Clinical Center, and in part by a grant from the KRIIB Research Initiative Program (Korean Biomedical Scientist Fellowship Program), Korea Research Institute of Bioscience and Biotechnology, Republic of Korea. We thank NVIDIA for the K40 GPU donation.

References

- [1] Mesh: Medical subject headings. <https://www.nlm.nih.gov/mesh/meshhome.html>.
- [2] Open-i: An open access biomedical search engine. <https://openi.nlm.nih.gov>.
- [3] U. Avni, H. Greenspan, E. Konen, M. Sharon, and J. Goldberger. X-ray categorization and retrieval on the organ and pathology level, using patch-based visual words. *Medical Imaging, IEEE Transactions on*, 2011.
- [4] S. Bird, E. Klein, and E. Loper. *Natural language processing with Python*. ” O'Reilly Media, Inc.”, 2009.
- [5] H. Boussaid and I. Kokkinos. Fast and exact: Admm-based discriminative shape segmentation with loopy part models. 2014.
- [6] X. Chen and C. Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *CVPR*, 2015.
- [7] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [9] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Gated feedback recurrent neural networks. *arXiv preprint arXiv:1502.02367*, 2015.
- [10] S. J. Darmoni, L. F. Soualmia, C. Letord, M.-C. Jaulet, N. Griffon, B. Thirion, and A. Névéol. Improving information retrieval using medical subject headings concepts: a test case on rare and chronic diseases. *Journal of the Medical Library Association: JMLA*, 100(3):176, 2012.
- [11] D. Demner-Fushman, M. D. Kohli, M. B. Rosenman, S. E. Shooshan, L. Rodriguez, S. Antani, G. R. Thoma, and C. J. McDonald. Preparing a collection of radiology examinations for distribution and retrieval. *Journal of the American Medical Informatics Association*, page ocv080, 2015.
- [12] D. Demner-Fushman, S. E. Shooshan, L. Rodriguez, S. Antani, and G. R. Thoma. Annotation of chest radiology reports for indexing and retrieval. *Multimodal Retrieval in the Medical Domain (MRMD) 2015*.
- [13] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.
- [14] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.
- [15] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015.
- [16] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*. 2010.
- [17] Y. Feng and M. Lapata. How many words is a picture worth? automatic caption generation for news images. In *Proceedings of the 48th annual meeting of the Association for Computational Linguistics*, 2010.
- [18] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.
- [19] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013.
- [20] R. B. Haynes, N. Wilczynski, K. A. McKibbon, C. J. Walker, and J. C. Sinclair. Developing optimal search strategies for detecting clinically sound studies in medline. *Journal of the American Medical Informatics Association*, 1(6):447, 1994.
- [21] S. Hermann. Evaluation of scan-line optimization for 3d medical image registration. In *CVPR*, 2014.
- [22] W. Hersh and E. Voorhees. Trec genomics special issue overview. *Information Retrieval*, 12(1):1–15, 2009.
- [23] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- [24] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [25] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 2013.
- [26] J. Hofmanninger and G. Langs. Mapping visual features to semantic profiles for retrieval in medical imaging. In *CVPR*, 2015.
- [27] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [28] O. Irsoy and C. Cardie. Opinion mining with deep recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- [29] S. Jaeger, A. Karargyris, S. Candemir, L. Folio, J. Siegelman, F. Callaghan, Z. Xue, K. Palaniappan, R. K. Singh, S. Antani, et al. Automatic tuberculosis screening using chest radiographs. *Medical Imaging, IEEE Transactions on*, 2014.
- [30] R. Jozefowicz, W. Zaremba, and I. Sutskever. An empirical exploration of recurrent network architectures. In *ICML 2015*, 2015.
- [31] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.
- [32] A. Karpathy, J. Johnson, and F.-F. Li. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*, 2015.
- [33] G. Kim, S. Moon, and L. Sigal. Joint photo stream and blog post summarization and exploration. In *CVPR*, 2015.
- [34] G. Kim, S. Moon, and L. Sigal. Ranking and retrieval of image sequences from multiple paragraph queries. In *CVPR*, 2015.
- [35] G. Kim, L. Sigal, and E. P. Xing. Joint summarization of large-scale collections of web images and videos for story-line reconstruction. In *CVPR*, 2014.

- [36] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *Transactions of the Association for Computational Linguistics*, 2015.
- [37] B. Klein, G. Lev, G. Sadeh, and L. Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, 2015.
- [38] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. 2012.
- [39] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. Berg. Babytalk: Understanding and generating simple image descriptions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(12):2891–2903, 2013.
- [40] C. Ledig, W. Shi, W. Bai, and D. Rueckert. Patch-based evaluation of image segmentation. In *CVPR*, 2014.
- [41] M. Lin, Q. Chen, and S. Yan. Network in network. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.
- [42] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*. 2014.
- [43] M.-T. Luong, I. Sutskever, Q. V. Le, O. Vinyals, and W. Zaremba. Addressing the rare word problem in neural machine translation. In *Proceedings of ACL*, 2015.
- [44] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.
- [45] J. Melendez, B. van Ginneken, P. Maduskar, R. H. Philipsen, K. Reither, M. Breuninger, I. M. Adetifa, R. Maane, H. Ayles, C. Sanchez, et al. A novel multiple-instance learning-based approach to computer-aided detection of tuberculosis on chest x-rays. *Medical Imaging, IEEE Transactions on*, 2015.
- [46] T. A. Ngo and G. Carneiro. Fully automated non-rigid segmentation with distance regularized level set evolution initialized and constrained by deep-structured inference. In *CVPR*, 2014.
- [47] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002.
- [48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011.
- [49] C. Rupprecht, L. Peter, and N. Navab. Image segmentation in twenty questions. In *CVPR*, 2015.
- [50] Y. Shi, H.-I. Suk, Y. Gao, and D. Shen. Joint coupled-feature representation and coupled boosting for ad diagnosis. In *CVPR*, 2014.
- [51] H.-C. Shin, L. Lu, L. Kim, A. Seff, J. Yao, and R. M. Summers. Interleaved text/image deep mining on a very large-scale radiology database. In *CVPR*, 2015.
- [52] H.-C. Shin, L. Lu, L. Kim, A. Seff, J. Yao, and R. M. Summers. Interleaved text/image deep mining on a large-scale radiology database for automated image interpretation. *Journal of Machine Learning Research (JMLR)*, 2016.
- [53] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [54] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics*, 2014.
- [55] N. Subbanna, D. Precup, and T. Arbel. Iterative multilevel mrf leveraging context and voxel information for brain tumour segmentation in mri. In *CVPR*, 2014.
- [56] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS 2014*.
- [57] T. Syeda-Mahmood, R. Kumar, and C. Compas. Learning the correlation between images and disease labels using ambiguous learning. In *MICCAI*. 2015.
- [58] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. 2015.
- [59] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008.
- [60] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, and K. Saenko. Sequence to sequence-video to text. *ICCV*, 2015.
- [61] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.
- [62] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. 2015.
- [63] Z. Xue, D. You, S. Candemir, S. Jaeger, S. Antani, L. R. Long, and G. R. Thoma. Chest x-ray image view classification. In *Computer-Based Medical Systems (CBMS), 2015 IEEE 28th International Symposium on*, 2015.
- [64] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. *ICCV*, 2015.
- [65] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2014.

Appendices

A. More Examples of the Chest X-ray Dataset

More examples of chest x-ray image, report, and annotations available on OpenI [2] is shown in Figure 7.

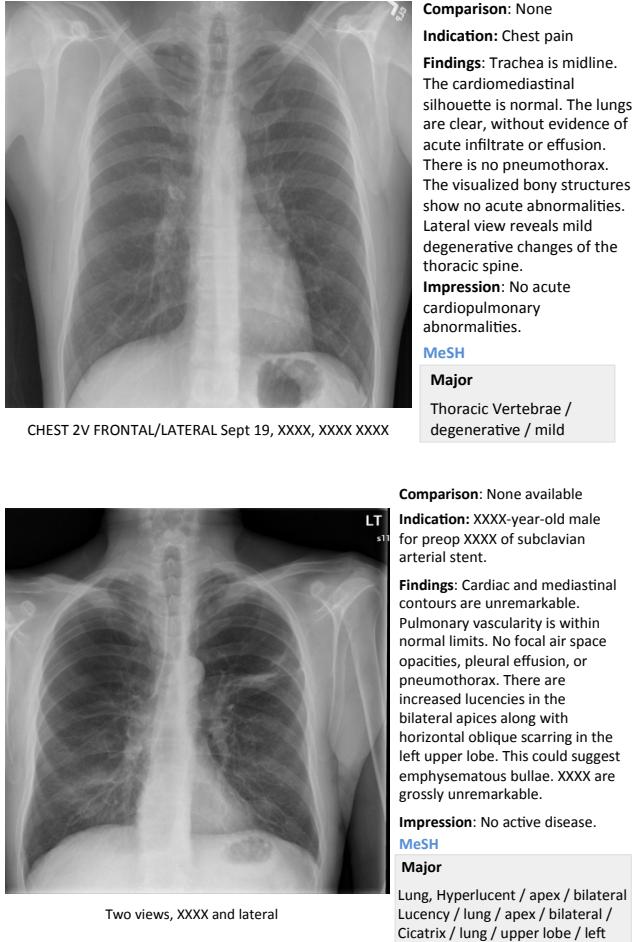


Figure 7. Examples of the OpenI [2] chest x-ray image, report, and annotations.

B. Visualizations of Joint Image/Text Context Vectors

Figure 8 shows the $h_{\text{im:text}}$ vectors of about fifty annotations describing *calcified granuloma* are projected onto two-dimensional planes via dimensionality reduction ($\mathbb{R}^{1 \times 1024} \rightarrow \mathbb{R}^{1 \times 2}$), using the t-SNE [59] implementation of [48]. Figure 9 shows the visualization for the annotations describing *opacity*. We can see that different annotations describing a disease are thereby separated into different categories by the $h_{\text{im:text}}$.

C. More Annotation Generation Examples

More annotation generation examples are provided in Figures 10 and 11. Overall, the system generates promising results on predicting disease (labels) and its context (attributes) in the images. However, rare disease cases are more difficult to detect. For example, the cases *pulmonary atelectasis*, *spondylosis*, and *density* (Figure 10), as well as *foreign bodies*, *atherosclerosis*, *costophrenic angle deformity* (Figure 11) are much rarer in the data than *calcified granuloma*, *cardiomegaly*, and all the frequent cases listed in Table 1 of the main paper.

Furthermore, the (left or right) location of the disease cannot be identified in a lateral view (obtained by scanning the patient from the side), as shown in Figure 11. Since our dataset contains a limited number of disease cases, we treat each x-ray image and report as a sample, and do not account for different views.

For future work, we plan to improve prediction accuracy by both (a) accounting for the different views, and (b) collecting a larger dataset to better account for rare diseases.

```

calcified_granuloma lung lower_lobe left multiple
calcified_granuloma lung upper_lobe left small
calcified_granuloma lung base right
calcified_granuloma lung hilum right
calcified_granuloma lung upper_lobe right multiple
calcified_granuloma lung base left
calcified_granuloma lung upper_lobe right

calcified_granuloma lung hilum right
calcified_granuloma lung lower_lobe right large
calcified_granuloma lung multiple
calcified_granuloma lung apex right small

calcified_granuloma lung hilum bilateral calcified_granuloma lung hilum right large
calcified_granuloma lung multiple
calcified_granuloma lung hilum right large
calcified_granuloma lung upper_lobe right calcified_granuloma spine degenerative
calcified_granuloma lung upper_lobe right calcified_granuloma scattered spine degenerative
calcified_granuloma multiple thoracic_vertebrae degenerative

calcified_granuloma lung lower_lobe left calcinosis lung hilum lymph_nodes multiple
calcified_granuloma lung lower_lobe right calcinosis lung hilum lymph_nodes multiple
calcified_granuloma lung bilateral multiple
calcified_granuloma multiple

calcified_granuloma lung lower_lobe left calcinosis lung hilum lymph_nodes multiple
calcified_granuloma lung lower_lobe left thoracic_vertebrae degenerative chronic
calcified_granuloma lung lower_lobe right calcified_granuloma lung lingula
calcified_granuloma lung middle_lobe right
calcified_granuloma lung left calcinosis right

```

```

calcified_granuloma lung right small
calcified_granuloma lung hilum bilateral multiple large calcinosis lung hilum lymph_nodes bilateral multiple large
calcified_granuloma lung upper_lobe right small

```

```

calcified_granuloma
calcified_granuloma
calcified_granuloma

```

Figure 8. Visualization of the $h_{im:text}$ of about 50 samples of the annotations describing the disease *calcified granuloma* on 2D plane.

```

opacity lung upper_lobe hilum left pneumonia upper_lobe left

opacity lung lower_lobe left pulmonary_atelectasis lower_lobe left

opacity lung base right pulmonary_atelectasis base right
opacity lung upper_lobe hilum left pneumonia upper_lobe left

opacity lung base left streaky pulmonary_atelectasis upper_lobe right
opacity lung middle_lobe right opacity lung middle_lobe right
opacity lung lingula pulmonary_atelectasis upper_lobe right
opacity lung lower_lobe right pulmonary_atelectasis lower_lobe right
opacity lung base left mild pulmonary_atelectasis base left

opacity lung base bilateral patchy streaky
opacity lung upper_lobe right focal pneumonia upper_lobe right
opacity lung upper_lobe hilum left pneumonia upper_lobe left
opacity lung upper_lobe right pneumonia upper_lobe right
opacity lung lower_lobe left focal pulmonary_atelectasis base left focal
opacity lung lower_lobe left pulmonary_atelectasis lower_lobe left
opacity lung ribs left anterior round
opacity lung upper_lobe right round
opacity lung lower_lobe right bilaterally diffuse reticular round severe
opacity lung lower_lobe bilateral interstitial
opacity lung hilum left interstitial
opacity thorax opacity lung base left pulmonary_atelectasis base right

opacity lung lower_lobe left pulmonary_atelectasis base left focal
opacity lung base bilateral pulmonary_atelectasis base bilateral
opacity lung base bilateral pulmonary_atelectasis base bilateral
opacity lung lower_lobe right streaky pulmonary_atelectasis lower_lobe right
opacity lung base bilateral pulmonary_atelectasis base bilateral
opacity lung base bilateral patchy streaky
opacity lung base bilateral
opacity lung lower_lobe bilateral interstitial

opacity lung lower_lobe bilateral
opacity lung bilaterally interstitial diffuse

opacity lung lower_lobe right pulmonary_atelectasis lower_lobe right

```

Figure 9. Visualization of the $h_{im:text}$ of about 50 samples of the annotations describing the disease *opacity* on 2D plane.

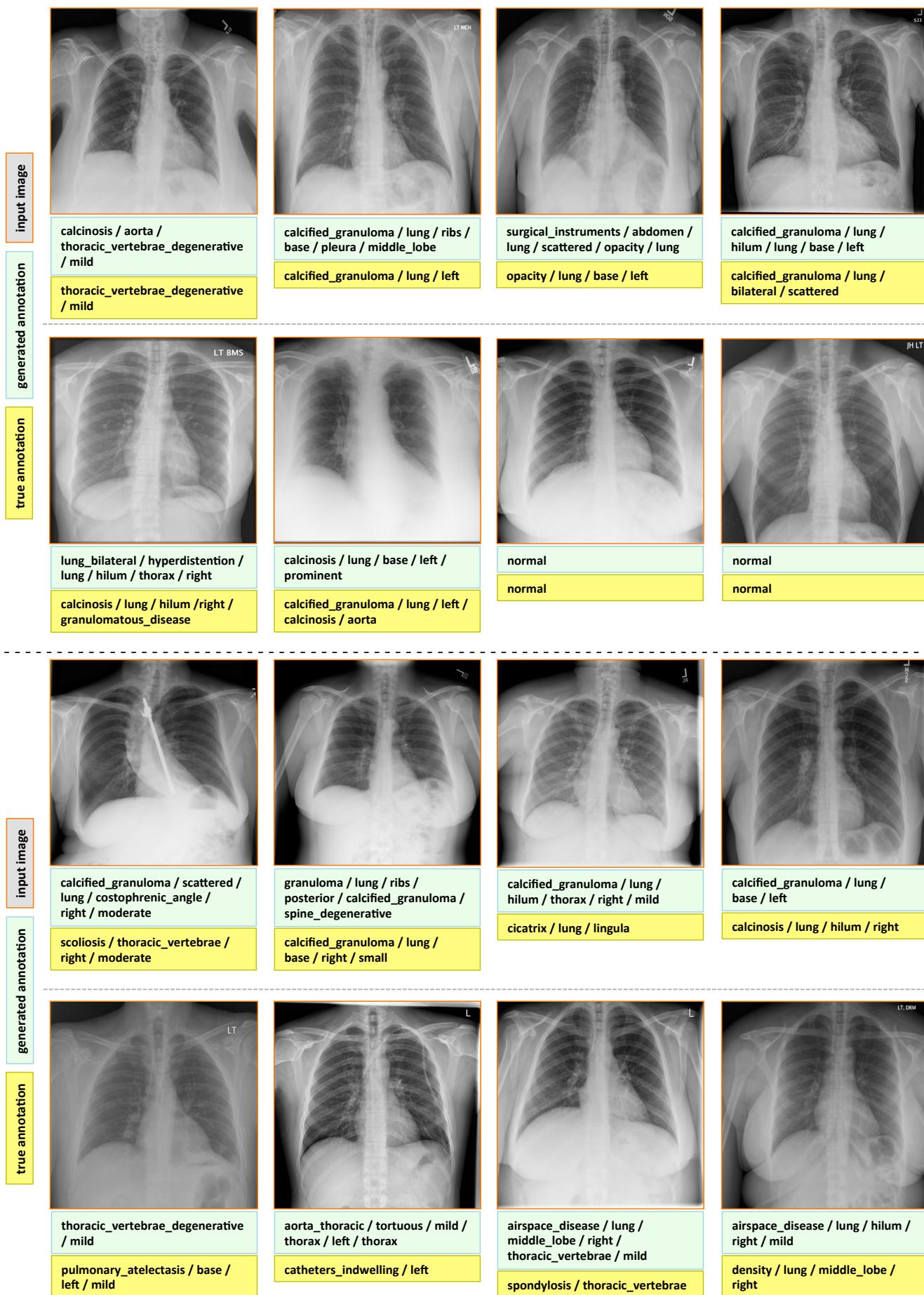


Figure 10. More examples of annotation generations (light green box) compared to true annotations (yellow box) for input images in the test set. Rare disease cases are more difficult to detect.

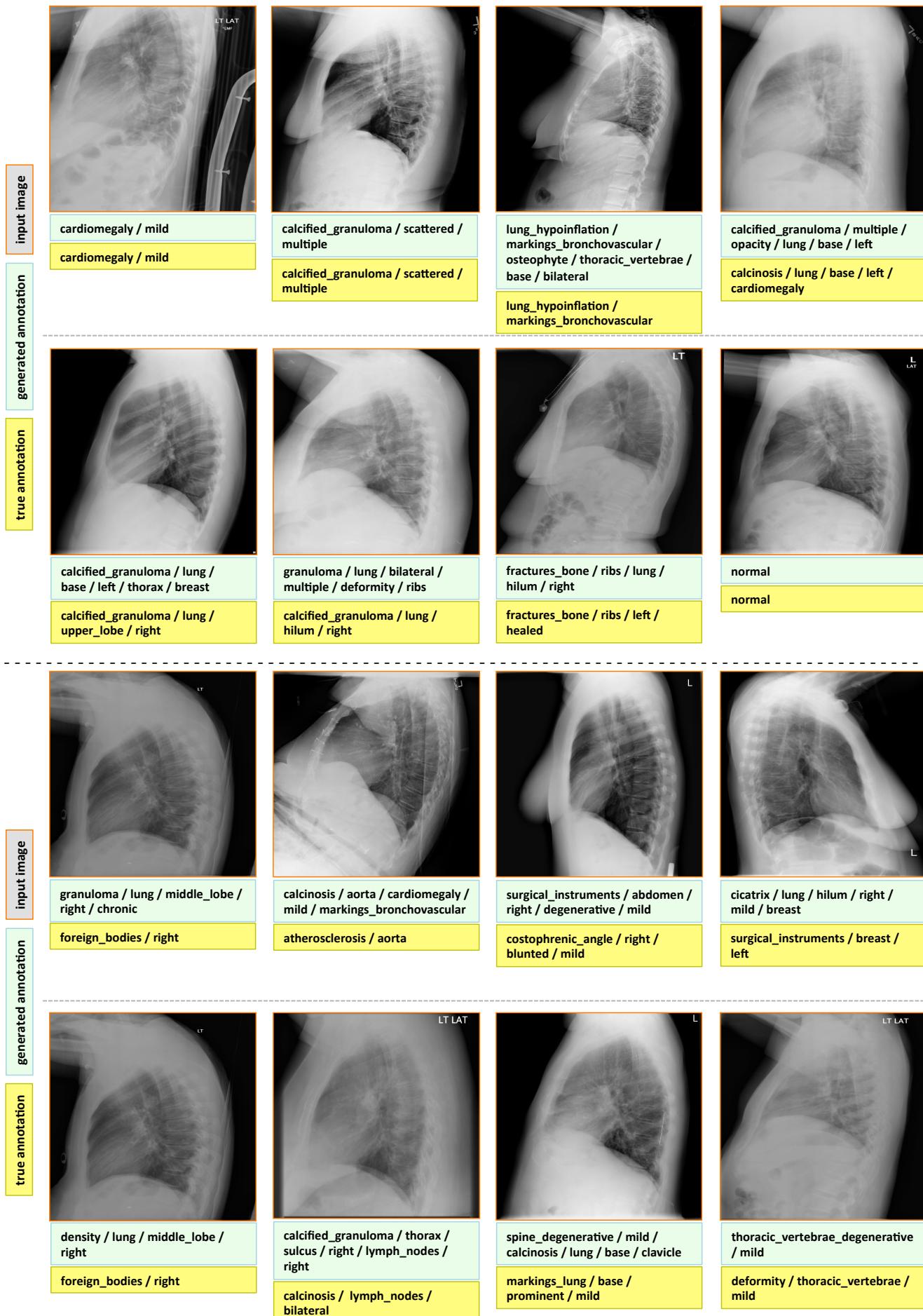


Figure 11. More examples of annotation generations (light green box) compared to true annotations (yellow box) for input images in the test set. In lateral (side) views, the (left or right) location of the disease can no longer be identified.